

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет
Кафедра автоматизації теплоенергетичних процесів

«На правах рукопису»
УДК 628.1+628.2

«До захисту допущено»
В.о.авідувача кафедри
_____ В.А.Волощук
(підпис)

«__»_____2019р.

Магістерська дисертація

На здобуття ступеня магістра

зі спеціальності 151 «Автоматизації та комп'ютерно-інтегровані технології»

на тему: Автоматизація міської системи водопостачання і водовідведення

Виконав: студент 2 курсу, групи ТА-81мп
(шифр групи)

_____ Лисак Дмитрій Юрійович _____
(Прізвище, ім'я, по-батькові) (підпис)

Науковий керівник _____ Асистент Гікало П.В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ Директор КП «Міськводоканал» СМР Сагач А.Г. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з
праць інших авторів без
відповідних посилань
Студент _____
(підпис)

Київ-2019

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Теплоенергетичний

Кафедра Автоматизації теплоенергетичних процесів

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

(підпис) В.А.Волощук
(ініціали, прізвище)

« ____ » _____ 2019р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Лисака Дмитрія Юрійовича
(група, прізвище, ім'я, по-батькові)

1.Тема дисертації: Автоматизація міської системи водопостачання і водовідведення

Науковий керівник дисертації: Асистент Гікало П.В.

затверджені наказом по університету від «4» листопада 2019р. № 3812-с

2.Термін подання студентів дисертації «10» грудня 2019р.

3.Об'єкт дослідження: Централізоване керування і збір даних з систем водопостачання і водовідведення

4.Предмет дослідження: Створення системи автоматичного керування міською системою водопостачання і водовідведення

5.Перелік завдань, які потрібно розробити: огляд структури системи водопостачання і водовідведення, порівняльний аналіз програмно-технічного комплексу для створення системи автоматизації, формування вимог до системи автоматизації, реалізація створеної системи, розробка стартап проекту

6.Орієнтовний перелік ілюстративного матеріалу: __технічна схема водопостачання міста Суми, архітектура системи автоматизації, функціональні одиниці системи автоматизації,знімки графічного інтерфейсу користувача_____

7.Орієнтовний перелік публікацій: Лисак Д.Ю., Гікало П.В. Система енергоефективного приватного будинку// Матеріали XVII Міжнародної науково-практичної конференції молодих вчених та студентів м. Київ, 23-26 квітня 2019 року, м.Київ, Україна. - ТОМ 2. С. 25.

8.Дата видачі завдання «4» вересня 2018р.

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Вивчення і аналіз задачі	02.10.18-09.10.18	
2.	Розробка архітектури та вибір програмно-технічного комплексу	10.10.18-24.12.18	
3.	Реалізація системи	10.02.19-27.10.19	
4.	Проходження переддипломної практики	02.09.2019-27.10.2019	
5.	Оформлення пояснювальної записки	06.09.19-15.11.19	
6.	Передзахист	9.12.19	
7.	Захист	19.12.19	

Студент

(підпис)

Д.Ю. Лисак
(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

П.В. Гікало
(ініціали, прізвище)

*Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Обсяг пояснювальної записки – 62 сторінок.

Пояснювальна записка містить 24 малюнків, 17 таблиць.

Ключові слова – АВТОМАТИЗАЦІЯ, ВОДОПОСТАЧАННЯ, ВОДОВІДВЕДЕННЯ, ВОДОКАНАЛ, ВНС, ПНС, КНС, MAJORDOMO, OPENSADA, RASPBERRY PI, ОБЕН, OPENWRT, VPN, ВЕБ-СЕРВЕР, АРАСНЕ, MODBUS RTU, MODBUS TCP.

Об'єкт дослідження – система водопостачання і водовідведення міста.

Мета роботи – розробка і реалізація дистанційної системи керування і контролю параметрів для системи водопостачання і водовідведення міста.

Завдання, які вирішуються для досягнення поставленої мети:

1. Аналіз структури системи водопостачання і водовідведення
2. Формулювання проблематики створення розподіленої системи керування та контролю параметрів
3. Аналіз існуючих технологій, програмного забезпечення та технічних засобів автоматизації
4. Вибір структури і складових системи автоматизації
5. Реалізація системи автоматизації

ABSTRACT

The volume of explanatory note is 62 pages.

Explanatory note contains 24 drawings, 17 tables.

Keywords – AUTOMATION, WATER SUPPLY, SANITATION, DWP, WATER PUMP STATION, OVERPRESSURE STATION, SEWER PUMPING STATION, MAJORDOMO, OPENSADA, RASPBERRY PI, OWEN, OPENWRT, VPN, WEB-SERVER, APACHE, MODBUS RTU, MODBUS TCP.

Object of research – city water supply and sanitation system.

The purpose of the work – development and implementation of a remote control and data control system for the city`s water supply and sanitation system.

Tasks that are solved to achieve the goal:

1. Analysis of the structure of water supply and sanitation system
2. Formulation of problems of creation of the distributed system of management and control of parameters.
3. Analysis of existing technologies, software, and automation tools.
4. Choosing the structure and components of the automation system.
5. Implementation of the automation system

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ СИМВОЛІВ І ТЕРМІНІВ	6
ВСТУП	7
1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ ВОДОПОСТАЧАННЯ І ВОДОВІДВЕДЕННЯ НА ПРИКЛАДІ ВОДОКАНАЛА МІСТА СУМИ	9
1.1 Аналіз сучасного стану технологічного забезпечення водоканалу в місті Суми	9
1.2 Опис структури водоканалу в м.Суми	10
1.3 Загальна постановка задачі.....	11
2 ОПИС СИСТЕМИ АВТОМАТИЗАЦІЇ.....	12
2.1 Призначення і функції створюваної системи	12
2.2 Вимоги до реалізації системи	13
2.3 Огляд засобів автоматизації.....	13
2.3.1 Протоколи і інтерфейси передачі даних	13
2.3.2 Програмно-логічні контролери і комп'ютери	15
2.3.3 Операційне і програмне забезпечення	16
2.3.4 Захист даних	18
2.4 Висновки	19
3 РОЗРОБКА СИСТЕМИ АВТОМАТИЗАЦІЇ ВОДОПОСТАЧАННЯ І ВОДОВІДВЕДЕННЯ МІСТА	21
3.1 Опис нижнього рівня автоматизації.....	21
3.2 Опис серверної частини	26
3.3 Опис верхнього рівня автоматизації.....	34
4 РОЗРОБКА СТАРТАП-ПРОЕКТУ ПО АВТОМАТИЗАЦІЇ СИСТЕМИ ВОДОПОСТАЧАННЯ І ВОДОВІДВЕДЕННЯ МІСТА	40
4.1 Основні положення ідеї.....	40
4.2 Опис області застосування проекту.....	40
4.3 Основні переваги для замовників.....	40
4.4 Аналіз конкурентів та порівняння системи з існуючими	41
4.5 Технологічний аудит ідеї проекту	43
4.6 Аналіз ринкових можливостей для запуску стартап-проекту.....	44
4.7 Розробка маркетингової компанії стартап-проекту.....	53
ВИСНОВОК	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
Додаток 1.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ СИМВОЛІВ І ТЕРМІНІВ

ПЛК	Програмований логічний контролер
SCADA	Диспетчерське керування та збір даних
DCS	Розподілена система керування
GSM	Глобальна система мобільно зв'язку
ПНС	Підвищувальна насосна станція, що необхідна для подачі води в багатоповерхові будинки
ВНС	Водонасосна станція, що включає в себе свердловини і станції другого підйому для подачі води в місто
КНС	Каналізаційна насосна станція, що перекачає стічні води до станції біологічної очистки
VPN	Віртуальна приватна мережа, що дозволяє об'єднати пристрої з усього світу в локальну мережу
SSH	Безпечна оболочка, що використовується для віддаленого керування комп'ютером
MPPE	Шифрування, створене для з'єднання типу точка-точка
MS-CHAP	Протокол перевірки автентичності підключення для пристроїв, що входять в приватну мережу

ВСТУП

Наразі мешканці будь-якого міста не можуть уявити своє існування без систем водопостачання і водовідведення. Тому для комунальних підприємств вкрай важливими є безвідмовність роботи системи і можливість якомога швидше та ефективно реагувати на нештатні ситуації, що можуть виникнути. Для забезпечення такого рівня послуг актуальним для міських водоканалів є впровадження систем автоматизації. Пристрої керування та збору інформації дозволяють отримувати данні про об'єкти систем водопостачання і водовідведення в реальному часі і можливість їх дистанційного керування, можливість аналізу роботи об'єктів системи, прийняття мір для підвищення ефективності використання ресурсів, і як наслідок збільшення прибутковості підприємства. Деякі виробники засобів автоматизації пропонують готові рішення для реалізації такого проекту, але впровадження даних рішень несуть за собою необхідність повної заміни встановленого обладнання, що є занадто капіталозатратним для комунального підприємства. Тому проблематика створення системи контролю і керування водопостачанням і водовідведенням полягає у поєднанні різноманітного обладнання, що знаходиться на великій відстані в єдину систему. Метою представленого проекту є проектування і розробка системи автоматизації водопостачання і водовідведення міста, що дозволить отримувати інформацію про об'єкти в реальному часі, виконувати керуючий вплив, аналізувати аварійні ситуації та ефективність роботи системи в цілому.

Завдання що вирішуються для досягнення поставленої мети:

1. Вивчення і аналіз існуючих методів побудови систем автоматизації водопостачання і водовідведення;
2. Розробка структури системи автоматизації;
3. Вибір приладів та програмного забезпечення;

4. Реалізація необхідних функцій на обраних приладах та програмному забезпеченні.

В якості об'єкта автоматизації в даній роботі буде розглянуто міську систему водопостачання і водовідведення міста Суми. Предметом цього проекту буде розглянута сучасна гнучка система автоматизації, що дає змогу збору інформації і керування різноманітними об'єктами автоматизації без необхідності зміни встановленого обладнання. На практиці це дозволить при мінімальних витратах реалізувати систему автоматизації, що наразі є єдиною можливою альтернативою для більшості міст України.

В даній роботі на захист виносяться наступні наукові положення і результати досліджень:

1. Теоретичний аналіз проблеми автоматизації системи водопостачання і водовідведення міста.
2. Комплексне рішення, що дозволяє реалізувати систему автоматизації без необхідності заміни існуючого обладнання.
3. Заходи по забезпеченню надійності роботи системи автоматизації.

Особистий вклад автора в дослідження полягає розробці структури системи автоматизації, виборі і обґрунтуванні використаних засобів автоматизації і програмного забезпечення і реалізації даної системи на прикладі системи водопостачання і водовідведення міста Суми.

1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПРОБЛЕМИ АВТОМАТИЗАЦІЇ ВОДОПОСТАЧАННЯ І ВОДОВІДВЕДЕННЯ НА ПРИКЛАДІ ВОДОКАНАЛА МІСТА СУМИ

1.1 Аналіз сучасного стану технологічного забезпечення водоканалу в місті Суми

Близько 70% технологічного обладнання, що використовується для забезпечення роботи водоканалу в місті Суми є фізично зношеним, а функціонал цього обладнання не дає змоги його інтеграції в систему автоматизації. Майже 90% свердловин, 40% насосів другого підйому та 60% каналізаційних насосів працюють на застарілих насосах з контакторами прямого пуску, що негативно відображається як на кількості споживаної електроенергії, так і зниженні середнього часу напрацювання на відмову. Велика кількість контрольно-вимірювальних приладів, таких як рівнеміри, витратоміри чи датчики тиску не мають уніфікованих вихідних сигналів і не дають змогу збору інформації. Трубопроводи знаходяться у аварійному стані.

Такий стан обладнання приводить до негативних наслідків:

- Втрата води в трубопроводах, що призводить до втрати коштів;
- Надлишковий тиск вночі в нерегульованих ділянках трубопроводів, що веде до підвищення можливості пориву труб.
- Неефективне використання електроенергії;
- Надлишковий знос обладнання і збільшення кількості експлуатаційних витрат на ремонт обладнання.

В теперішній час збір інформації з водозаборів і каналізаційних насосних станцій відбувається за допомогою працівників водоканалу, що ведуть журнали обліку, а керування об'єктами відбувається диспетчером у телефонному режимі, що призводить до унеможливлення відображення цілісної картини стану водопостачання і водовідведення міста та затрудняє можливість прийняття

правильних рішень. Всі ці фактори негативно впливають на роботу системи в цілому, оскільки не дають можливості аналізувати поточні данні, підбирати найбільш економічно доцільні режими роботи та вчасно реагувати на аварійні ситуації, що в результаті веде до зменшення прибутку комунального підприємства.

1.2 Опис структури водоканалу в м.Суми

Наразі в околицях місті знаходиться шість водозаборів підземних вод, з яких вода потрапляє в місто. У склад водозаборів входить сумарно 67 свердловин, 6 насосних станцій другого підйому та 16 резервуарів чистої води сумарною ємністю ~80000 м³. Через особливості ландшафту для потреб багатоповерхових будинків в мікрорайонах міста встановлено 7 підвищувальних насосних станцій. За водовідведення міста відповідають 19 каналізаційних насосних станцій та станція біологічної очистки стічних вод.

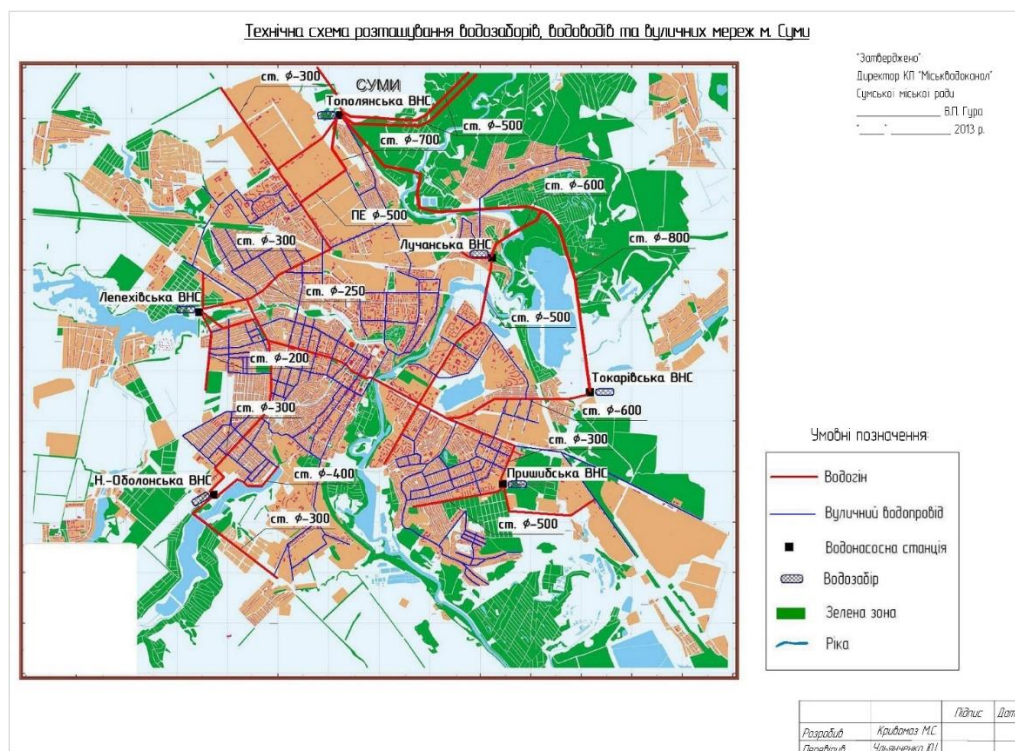


Рис.1 Технічна схема розташування водозаборів, водоводів та вуличних мереж м.Суми

1.3 Загальна постановка задачі

Задачею даної дисертації є розробка системи автоматизації міської системи водопостачання і водовідведення. Для цього буде виконано наступні підзадачі:

1. Аналіз структури, призначення і функціоналу об'єкту автоматизації.
2. Опис вимог що висовуються до системи автоматизації.
3. Аналіз можливих варіантів реалізації системи автоматизації.
4. Обґрунтування технічних засобів та структури системи автоматизації.
5. Реалізація програмного забезпечення для системи автоматизації.
6. Аналіз функціоналу і працездатності готової системи автоматизації.

2 ОПИС СИСТЕМИ АВТОМАТИЗАЦІЇ

2.1 Призначення і функції створюваної системи

Призначенням створюваної системи автоматизації є забезпечення можливості дистанційного збору інформації з засобів автоматизації та можливість здійснення керуючого впливу на них.

Основними функціями системи автоматизації є збір та запис в архів поточних значень з засобів автоматизації і контрольно-вимірювальних приладів, що встановлені в свердловинах, станціях другого і третього підйому, а також каналізаційних насосних станціях. Крім цього система має доступним чином представляти всю зібрану інформації в вигляді поточних значень на певному об'єкті, або у вигляді трендів на основі попередніх значень, що зберігаються в архівах. Також в вимоги до створення системи автоматизації входить ведення журналу подій та журналу аварій, що дають змогу аналізувати поточний стан системи та діагностувати причини аварій на основі попередніх подій. При виникненні аварій система має коректним чином відреагувати на аварійну ситуацію і виконати все можливе щоб усунути або мінімізувати можливі наслідки, а також привернути увагу оператора і дати максимально інформативне сповіщення про ситуацію, що виникла. Реалізація даних функцій дозволяє отримати цілісну картину поточного стану об'єкту автоматизації, аналізувати данні для оптимізації подальшої роботи, та вчасно реагувати на зміни в роботі об'єкта автоматизації.

Крім цього система автоматизації дозволяє дистанційно створювати керуючий вплив на засоби автоматизації, що встановлені на об'єктах. Реалізація даної функції дозволяє централізовано керувати об'єктами автоматизації, що дає змогу швидкого реагування на зміну рівня розбору в години підвищеного навантаження та зменшувати потужність системи в час низького розбору, що в цілому веде до зменшення затрат на електроенергію, та збільшення часу

напрацювання на відмову всіх встановлених засобів автоматизації і запірних арматур.

2.2 Вимоги до реалізації системи

Результатом створення системи автоматичного керування і збору інформації має бути універсальне рішення, що задовольняє наступним вимогам:

- Система має працювати безперервно;
- Кожен з об'єктів автоматизації має гарантовано з'являтися на зв'язку в проміжку певного часу, що визначений замовником, але не частіше, ніж це дозволяє пропускна здатність вибраного каналу зв'язку;
- Всі данні отримані з об'єктів мають бути збережені в базі даних в доступному вигляді і зберігатись не менше 3 місяців.
- Система має гарантовано відпрацьовувати всі аварійні алгоритми зупинки або аварійного режиму роботи об'єктів керування без необхідності зв'язку з сервером.
- Система має гарантовано відпрацьовувати всі команди оператора в незалежності від того, в якому стані знаходиться зв'язок чи контрольований об'єкт.
- Всі отримані дані мають бути представлені у вигляді зручного графічного інтерфейсу, задовольняти всім вимогам до UI/UX дизайну та оформлення SCADA-систем.

2.3 Огляд засобів автоматизації

2.3.1 Протоколи і інтерфейси передачі даних

Правильний вибір протоколів і інтерфейсів даних є важливим аспектом в розробці системи автоматизації, оскільки правильний вибір дозволить створити найоптимальнішу топологію всієї системи, налаштувати стабільну передачу інформації, а також в майбутньому завадити виникненню аварійних ситуацій.

З точки зору стабільності передачі даних, найнадійнішими інтерфейсами обміну можна назвати Ethernet і оптоволоконний кабель, але через специфіку територіального розміщення об'єктів автоматизації, що знаходяться на великій відстані один від одного, а іноді і на значній відстані від міста, можливість створення кабельного підключення стає важко досяжним, а іноді і неможливим. Тим не менше, при можливості підключення до дротової мережі, їх вибір буде найкращим.

Передача даних по Wi-Fi дозволяє передавати дані з великою швидкістю в середині локальної мережі, але обмежена типом підключення до мережі інтернет. Використання даної технології спричиняє велике навантаження на інтернет підключення, оскільки всі забрані дані передаються через один канал зв'язку і рекомендується для збору інформації з об'єктів, що знаходяться поруч і мають дротове підключення до інтернету.

Наступними в списку по стабільності передачі даних можна назвати бездротові мережі LoRa та NB-IoT, що більш обмежені по швидкості передачі пакетів, але дозволяють підтримувати стабільний зв'язок. Наразі данні системи проходять тестування в місті Києві, і поки є неможливим їх реалізація в невеликих містах.

Останньою в списку інтерфейсів передачі даних є технологія GSM зв'язку, що значно поступається всім іншим по надійності передачі сигналу, але в умовах реалізації розподіленої системи автоматизації в більшості випадків є єдиним доступним рішенням, яким наразі і є даний проект. Тому як основу даного проекту було обрано використання інтерфейсу GSM зв'язку.

Протокол обміну інформації – мова спілкування між пристроями в системі автоматизації. Провідні виробники засобів автоматизації розробили свої протоколи обміну даних, що дозволяють більш лаконічно і надійно передавати інформації між пристроями в системі. Але ці протоколи обміну є пропрієтарними і дозволяють спілкуватися між собою лише засобам автоматизації створених

одним виробником. Оскільки в системі засоби автоматизації представлені продукцією різних виробників, то протоколом обміну даними було обрано ModBus RTU для реалізації зв'язку між приладами всередині одного об'єкту автоматизації. Для зв'язку об'єктів з системою диспетчерського керування та збору даних найкращим варіантом наразі є використання протоколу MQTT, що дозволяє підтримувати стабільний обмін даними навіть в умовах нестабільного зв'язку та низької пропускної здатності каналу передачі. Крім цього, при використанні другого рівня QoS, має гарантовану доставку пакетів, що є дуже важливою функцією в системі керування. Але через вимоги замовника даний проект буде реалізовано з використанням протоколу обміну ModBus TCP.

2.3.2 Програмно-логічні контролери і комп'ютери

Наразі на ринку представлена велика кількість засобів автоматизації від різноманітних виробників, що відрізняються вартістю, середовищем розробки, надійністю, обчислювальною потужністю, та додатковим функціоналом. Але кожен з них має задовольняти вимогам, що на нього покладені, бо помилка системи автоматизації можуть призвести до виходу з ладу певної ланки, чи всієї системи в цілому, та виникнення аварійних незворотних ситуацій.

При виборі засобів автоматизації для даної системи потрібно враховувати велику відстань між різними об'єктами, тому важливим критерієм є підтримка GSM зв'язку. Такі компанії як Siemens, Schneider Electric, Phoenix Contact мають набірні модулі GSM зв'язку. Але наразі закупівля рішень по автоматизації, що базуються на обладнанні даних виробників є занадто дорогою для комунальних підприємств України, тому реалізація системи автоматизації була побудована на обладнанні компанії ОВЕН.

Для реалізації підтримки GSM зв'язку, обробки, збереження і передачі інформації отриманої з контролера в якості шлюзу даних було використано одноплатний мікрокомп'ютер Raspberry Pi. Використання цього

мікрокомп'ютера дозволяє вести резервну локальну базу даних, що незалежна від зв'язку з сервером і дозволяє відслідковувати зміну параметрів з меншою дискретністю. Також за допомогою Raspberry Pi виконуються функції обробки і перетворення інформації, що ведуть до зменшення розміру даних, що передаються та функції перетворення запитів з протоколу ModBus RTU в ModBus TCP і назад. Крім цього використання Raspberry Pi дозволяє дистанційно змінювати налаштування контролера і засобів автоматизації, так як плавні пуски чи частотні перетворювачі, що в умовах DCS та постійно змінюваних засобів автоматизації є досить важливою функцією.

2.3.3 Операційне і програмне забезпечення

Велику роль у розробці системи автоматизації відіграє обране операційне і програмне забезпечення, що повинно мати відповідний функціонал, який має повністю задовільнити вимоги до кінцевого продукту та максимально стабільно відпрацьовувати свої функції.

В якості операційного забезпечення було обрано операційну систему Linux. Переваги операційної системи Linux в порівнянні з операційною системою Windows полягають в першу чергу в стабільності роботи, захищеності даних, що зберігаються на комп'ютері, гнучкістю системи, а також, немаловажливим є безкоштовність даного операційно забезпечення, що дозволяє заощадити гроші замовника.

Для збору і обробки інформації необхідним є вибір відповідного програмного забезпечення, що дозволило б стабільно збирати і передавати данні, реагувати на зміни в системі і правильно на них реагувати. Наразі системи збору і обробки інформації, що виробляються провідними компаніями, мають обмежені можливості, оскільки дозволяють в повному об'ємі працювати лише з обладнанням цієї ж компанії, і крім цього потребують додатковим витрат на закупівлю ліцензій. Інші ж програми мають обмежений функціонал, що не

дозволяють реалізувати повний спектр необхідних функцій для системи автоматизації. В якості системи збору і обробки даних було обрано програмне забезпечення OpenSCADA. Дане програмне забезпечення не є пропрієтарним, що дозволяє працювати з обладнанням різноманітних виробників, використовуючи власні бібліотеки та функції. OpenSCADA є середовищем розробки і дає можливості самостійно створювати необхідні додаткові функції. Це програмне забезпечення встановлюється на мікрокомп'ютери Raspberry Pi та головний сервер. За допомогою реалізованих модулів збору інформації, обробки і можливості виступати шлюзом даних, дане програмне забезпечення дозволяє збирати інформацію з локальних пристроїв, обробляти її та передавати до сервера, де вона в структурованому вигляді записується в базу даних.

Збереження інформації в базах даних дозволяють отримувати ретроспективу значень параметрів, будувати графіки, аналізувати їх та оптимізувати подальшу роботу системи. Найпопулярнішими СУБД є Oracle і MySQL. Але в даному проекті було обрано СУБД MariaDB, що за рахунок оптимізованої структури даних дозволяє одночасно більш ефективно опрацьовувати запити і за рахунок колонкового зберігання більш ефективно опрацьовує великі запити, що необхідно при побудові графіків. Також при виборі вигляду збереження інформації важливим є обрання структури таблиць. Найкращими структурами є InnoDB та MyISAM. Структура таблиць MyISAM є більш оптимізованою, і дозволяє опрацьовувати великі запити більш ефективно. В той же час InnoDB є більш повільною, але за рахунок ведення логів таблиць, є більш надійною і в разі чого, здатною сама себе відновити. Поєднання оптимізованої СУБД MariaDB і надійної структури таблиць InnoDB разом дають досить високу швидкість обробки запитів і надійність збереження даних.

Для створення користувацького інтерфейсу було обрано програмне забезпечення MajorDoMo. Перевагами даної програми є її безмежний функціонал, оскільки її можна вважати середовищем розробки, тобто програмне

забезпечення виступає лише посередником, і дозволяє за допомогою PHP, HTML, CSS та JavaScript організувати алгоритми зчитування інформації з бази даних, функції її обробки та можливість її відображення у зручному для користувача вигляді. Програма MajorDoMo знаходиться на веб-сервері Apache, що дозволяє відслідковувати стан об'єктів системи автоматизації з будь-якої точки світу, це є важливим для обслуговування системи, оскільки безліч спеціалістів в певних сферах можуть одночасно користуватись системою.

2.3.4 Захист даних

Так як система автоматизації дозволяє передавати керуючий вплив на об'єкти водопостачання і водовідведення міста, важливим є захист всієї системи від втручання і захист каналів від перехвату інформації, оскільки втручання злоумисників в роботу системи може призвести до виходу з ладу обладнання, припинення водопостачання міста, або створення екологічної катастрофи шляхом внесення змін в режим роботи КНС. Для цього необхідним є проведення заходів по захисту інформації.

Першим пунктом є створення VPN. Віртуальна приватна мережа дозволяє об'єднати всі об'єкти автоматизації і користувачів в єдину локальну мережу. Пакети, що передаються всередині віртуальної локальної мережі шифруються за допомогою ключів `trre-128`, а всі пристрої проходять автентифікацію за допомогою протоколу `MS-CHAP`, що захищають інформацію і роблять неможливим втручання сторонніх пристроїв в локальну мережу.

Для дистанційної роботи з сервером і його технічної підтримки виникає необхідність для відкриття доступу по `SSH`. Для захисту серверу від втручання з глобальної мережі використано алгоритм хешування `SHA256` та режим обміну ключами `RSA`, що робить неможливим втручання в роботу системи зі сторонніх пристроїв.

2.4 Висновки

В результаті огляду і аналізу засобів автоматизації для реалізації системи була обрана наступна структура ПТКЗА.

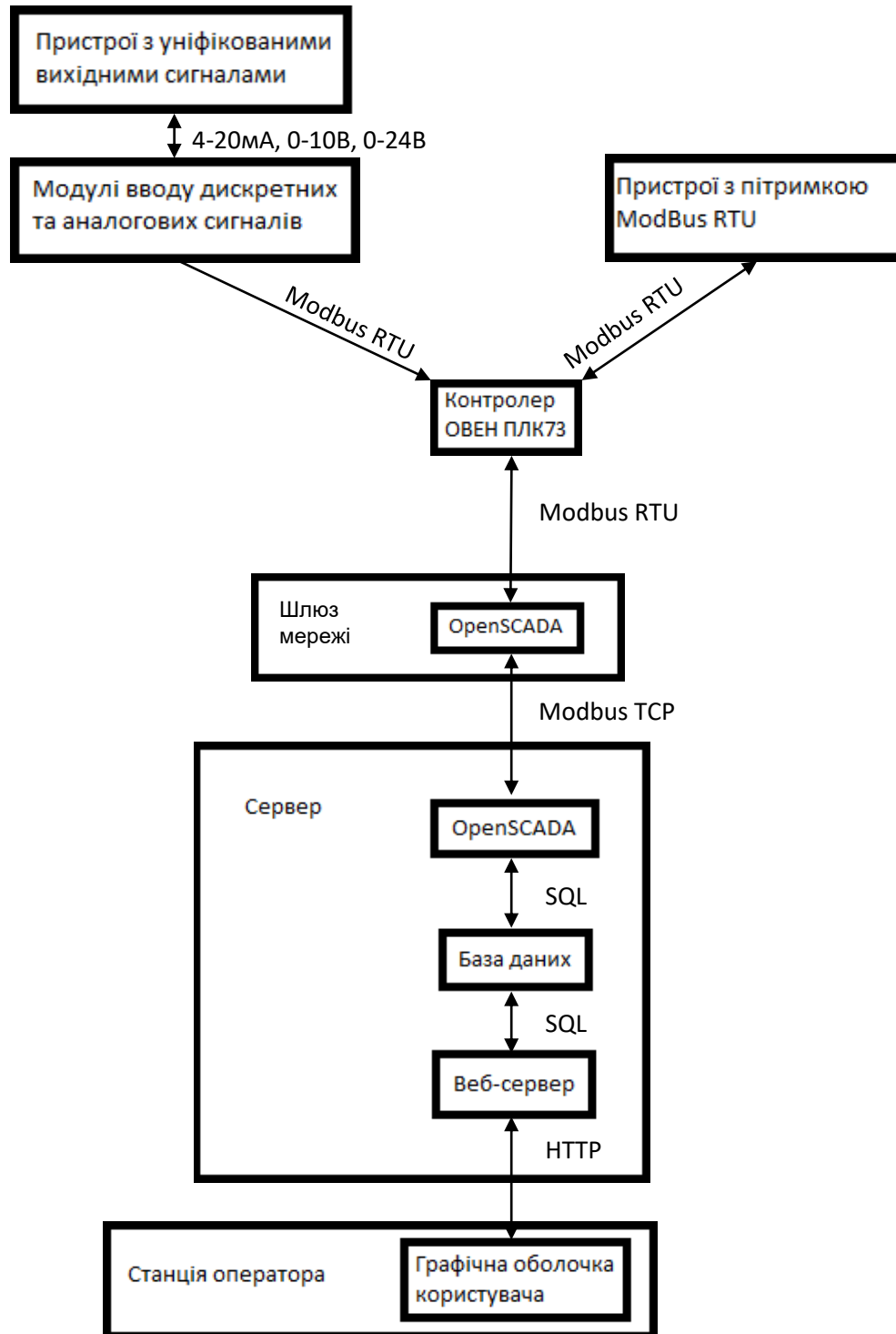


Рис. 2 Структурна схема ПТКЗА

Данна структура програмно-технічного комплексу засобів автоматизації є універсальним рішенням, оскільки дозволяє об'єднувати в одну систему обладнання різних виробників і з різними вихідними сигналами, що в реаліях сучасного стану технічного забезпечення водоканалів України є єдиним доступним рішенням для реалізації системи автоматизації.

Крім цього, перевагами такої структури є можливість віддаленого доступу до системи, що з врахуванням реалізації рівнів доступу до функції керування, робить систему автоматизації інструментом аналізу ефективності роботи об'єктів для спеціалістів різних областей.

3 РОЗРОБКА СИСТЕМИ АВТОМАТИЗАЦІЇ ВОДОПОСТАЧАННЯ І ВОДОВІДВЕДЕННЯ МІСТА

3.1 Опис нижнього рівня автоматизації

В якості головної функціональної одиниці нижнього рівня автоматизації було обрано контролер ОВЕН ПЛК73, функціями якого є збір інформації з пристроїв аналогового та дискретного вводу, та інших засобів автоматизації, відпрацювання головних циклів роботи системи, прийняття рішень про керуючий вплив в автоматичному режимі, у разі, якщо це дозволено оператором, а також відпрацювання алгоритмів аварійної зупинки та аварійної роботи системи. Дані контролери, незважаючи на низьку ціну, дозволяють реалізувати неперервну роботу системи, мають широкий функціонал і велику кількість модулів розширення, що дозволяє повністю реалізувати функції, що покладені на систему автоматизації. Головним недоліком контролерів ОВЕН є їх порівняно низька надійність, але через специфіку систем водопостачання, в яких помилка в роботі окремої складової має відносно низьку критичність як для роботи системи в цілому, так і для навколишнього середовища, цим недоліком можна знехтувати. Приклад програми для реалізації керування підвищувальною насосною станцією на контролері ОВЕН приведений в Додатку 1.



Рис. 3.1 Зовнішній вигляд контролера ОВЕН ПЛК73

Для реалізації розподіленої системи автоматизації важливим є обмін інформації між складовими системи і головним сервером, що дозволяє реалізувати відображення цілісної картини і можливості дистанційного керування. Для реалізації шлюзу даних між контролером і головним сервером в рамках цього проекту було використано одноплатний мікрокомп'ютер Raspberry Pi. На мікрокомп'ютер було встановлено операційне забезпечення OpenWRT, що є значно кращим за офіційний дистрибутив Raspbian, оскільки є більш оптимізованим, і має велику кількість вбудованих алгоритмів самодіагностування, що забезпечує безперервну роботу системи. Основними функціями Raspberry Pi є збір і передача інформації, що реалізується за допомогою встановлення окремих модулів програмного забезпечення OpenSCADA. Специфікою роботи з контролерами ОВЕН є те, що за один запит з контролера можливо зчитати лише один регістр інформації. Тому функції мікрокомп'ютера полягають у зчитуванні і накопиченні отриманої інформації, оптимізації отриманих даних для подальшої їх передачі, та забезпечення неперервного зв'язку з мережею інтернет та VPN-сервером.



Рис. 3.2 Зовнішній вигляд одноплатного мікрокомп'ютера Raspberry Pi

Розглянемо порядок передачі інформації від об'єкта автоматизації до графічного інтерфейсу користувача на прикладі свердловини.

```
Перечень атрибутів:  
R:260:r:LinearVoltage  
R:752:w:Control  
R_b0:257:r:Current1b0  
R_b1:257:r:Current1b1  
R_b2:257:r:Current1b2  
R_b3:257:r:Current1b3  
R_b4:257:r:Current1b4  
R_b5:257:r:Current1b5  
R_b6:257:r:Current1b6  
R_b7:257:r:Current1b7  
R_b8:257:r:Current1b8  
R_b8:256:r:SWb8  
R_b9:256:r:SWb9  
R_b10:256:r:SWb10  
R_b11:256:r:SWb11  
R_b12:256:r:SWb12  
R_b13:256:r:SWb13  
R_b0:259:r:Current3b0  
R_b1:259:r:Current3b1  
R_b2:259:r:Current3b2  
R_b3:259:r:Current3b3  
R_b4:259:r:Current3b4  
R_b5:259:r:Current3b5  
R_b6:259:r:Current3b6  
R_b7:259:r:Current3b7  
R_b8:259:r:Current3b8  
R_b0:256:r:Ready  
R_b1:256:r:Run  
R_b2:256:r:Trip  
R_b3:256:r:Warning  
R:3:r:NominalCurrent  
R:256:r:SW  
R:282:Error1  
R:283:Error2  
R:284:Error3
```

Рис. 3.3 Приклад реалізації модулю збору інформації ПЗ OpenSCADA на Raspberry Pi

На даному рисунку приведено перелік регістрів плавного пуску Schneider Electric з яких йде зчитування інформації за допомогою Raspberry Pi. В результаті ми отримуємо можливість керування плавним пуском шляхом зміни регістру керування та отримувати інформації про стан плавного пуску, його лінійну напругу, величину струму на фазах та пояснення останньої аварії, що виникла на плавному пускі.

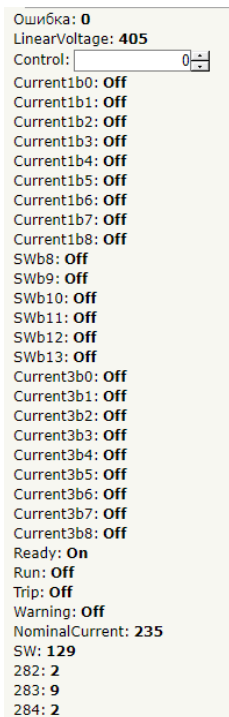


Рис. 3.4 Результат роботи модулю збору інформації ПЗ OpenSCADA на Raspberry Pi

```

C1 = ModBus.ATS22.ch1.Current1b8 * 256 + ModBus.ATS22.ch1.Current1b7 * 128 + ModBus.ATS22.ch1.Current1b6 * 64
+ ModBus.ATS22.ch1.Current1b5 * 32 + ModBus.ATS22.ch1.Current1b4 * 16 + ModBus.ATS22.ch1.Current1b3 * 8
+ ModBus.ATS22.ch1.Current1b2 * 4 + ModBus.ATS22.ch1.Current1b1 * 2 + ModBus.ATS22.ch1.Current1b0;
C2 = ModBus.ATS22.ch1.Current3b8 * 256 + ModBus.ATS22.ch1.Current3b7 * 128 + ModBus.ATS22.ch1.Current3b6 * 64
+ ModBus.ATS22.ch1.Current3b5 * 32 + ModBus.ATS22.ch1.Current3b4 * 16 + ModBus.ATS22.ch1.Current3b3 * 8
+ ModBus.ATS22.ch1.Current3b2 * 4 + ModBus.ATS22.ch1.Current3b1 * 2 + ModBus.ATS22.ch1.Current3b0;
if (C3 == 1) {ModBus.ATS22.ch1.Control = 11;}
if (C3 == 0) {ModBus.ATS22.ch1.Control = 0;}
if (C4 == 1) {ModBus.ATS22.ch1.Control = 8;}
if (ModBus.ATS22.ch1.Trip == 0) {C4 = 0;}

```

Рис. 3.4 Приклад функції обробки вхідних даних, реалізації керування та оптимізації вихідних даних в ПЗ OpenSCADA на Raspberry Pi

Реалізована функція перетворює отриману з плавного пуску інформації в більш зручний вигляд, шляхом переведення бітів в з регістру поточного струму в ціле число, та встановлює мітки (C3, C4), що в спрощують реалізацію керування. Шляхом зміни значення міток реалізується запуск і зупинка плавного пуску, а також дистанційне скидання поточної аварії.

```

Current1: ModBus.iStart.r_2.Current1 (+)
LinearVoltage2: ModBus.iStart.r_2.LinearVoltage (+)
Current3: ModBus.iStart.r_2.Current3 (+)
Start/Stop/Reset: ModBus.iStart.r_2.Control (+)
Start/Stop: JavaLikeCalc.RasschetNapryazheniy.param.C3 (+)
Reset: JavaLikeCalc.RasschetNapryazheniy.param.C4 (+)
DI1: ModBus.DI.ch1.B0 (+)
DI2: ModBus.DI.ch1.B1 (+)
DI3: ModBus.DI.ch1.B2 (+)
DI4: ModBus.DI.ch1.B3 (+)
DI5: ModBus.DI.ch1.B4 (+)
DI6: ModBus.DI.ch1.B5 (+)
DI7: ModBus.DI.ch1.B6 (+)
DI8: ModBus.DI.ch1.B7 (+)
DI9: ModBus.DI.ch1.B8 (+)
DI10: ModBus.DI.ch1.B9 (+)
DI11: ModBus.DI.ch1.B10 (+)
DI12: ModBus.DI.ch1.B11 (+)
DI13: ModBus.DI.ch1.B12 (+)
DI14: ModBus.DI.ch1.B13 (+)
DI15: ModBus.DI.ch1.B14 (+)
DI16: ModBus.DI.ch1.B15 (+)
AI1: ModBus.AI.ch1.AI1 (+)
AI2: ModBus.AI.ch1.AI2 (+)
AI3: ModBus.DI.ch1.Reg33
LinearVoltage1: ModBus.iStart.r_2
StatusWord: ModBus.iStart.r_2.SW (+)
Error1: ModBus.iStart.r_2.282 (+)
Error2: ModBus.iStart.r_2.283 (+)
Error3: ModBus.iStart.r_2.284 (+)
Count1: ModBus.DI.ch1.Count1
Count2: ModBus.DI.ch1.Count2
Count3: ModBus.DI.ch1.Count3
Count4: ModBus.DI.ch1.Count4
DI17: ModBus.DI.ch1.B16 (+)
DI18: ModBus.DI.ch1.B17 (+)
DI19: ModBus.DI.ch1.B18 (+)
DI20: ModBus.DI.ch1.B19 (+)
DI21: ModBus.DI.ch1.B20 (+)
DI22: ModBus.DI.ch1.B21 (+)
DI23: ModBus.DI.ch1.B22 (+)
DI24: ModBus.DI.ch1.B23 (+)
DI25: ModBus.DI.ch1.B24 (+)
DI26: ModBus.DI.ch1.B25 (+)
DI27: ModBus.DI.ch1.B26 (+)
DI28: ModBus.DI.ch1.B27 (+)
DI29: ModBus.DI.ch1.B28 (+)
DI30: ModBus.DI.ch1.B29 (+)
DI31: ModBus.DI.ch1.B30 (+)
DI32: ModBus.DI.ch1.B31 (+)
AI3: ModBus.AI.ch1.AI3 (+)
AI4: ModBus.AI.ch1.AI4 (+)
AI5: ModBus.AI.ch1.AI5 (+)
AI6: ModBus.AI.ch1.AI6 (+)
AI7: ModBus.AI.ch1.AI7 (+)
AI8: ModBus.AI.ch1.AI8 (+)

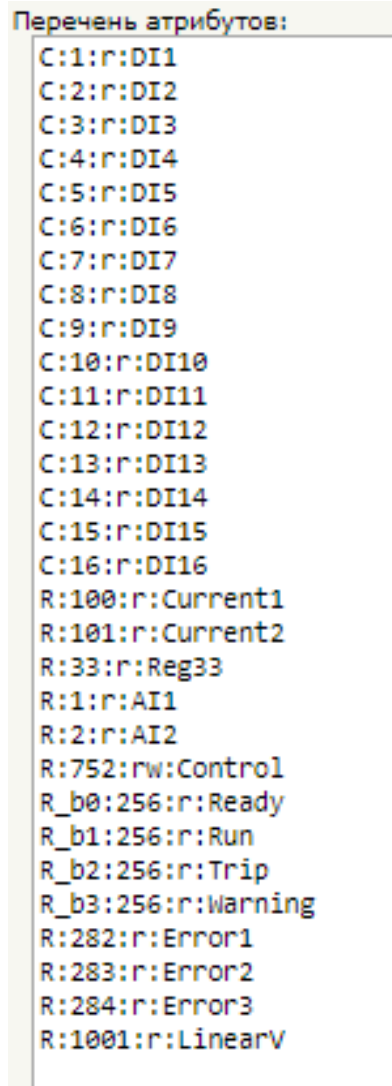
```

Рис. 3.5 Приклад реалізації шлюзу даних в ПЗ OpenSCADA на Raspberry Pi

Наступним кроком є реалізації шлюзу даних, що дозволяє за допомогою запитів, що надходять з серверу по протоколу ModBus TCP, зчитувати дані, що отримані з об'єкту автоматизації по протоколу ModBus RTU. На даному зображенні (Рис. 3.5) показано збір інформації з пристрою плавного пуску та модулів дискретного та аналогового вводу сигналів. Останні встановлені для зчитування інформації з датчиків тиску, датчиків проникнення, пожежі та затоплення.

3.2 Опис серверної частини

Робота серверної частини починається зі збору інформації з об'єктів. Протокол Modbus працює за архітектурою майстер - раб. Сервер в даній ієрархії виконує роль майстра, тим самим саме він визначає пріоритетність інформації та частоту опитування підконтрольних пристроїв.



Перечень атрибутов:

- C:1:r:DI1
- C:2:r:DI2
- C:3:r:DI3
- C:4:r:DI4
- C:5:r:DI5
- C:6:r:DI6
- C:7:r:DI7
- C:8:r:DI8
- C:9:r:DI9
- C:10:r:DI10
- C:11:r:DI11
- C:12:r:DI12
- C:13:r:DI13
- C:14:r:DI14
- C:15:r:DI15
- C:16:r:DI16
- R:100:r:Current1
- R:101:r:Current2
- R:33:r:Reg33
- R:1:r:AI1
- R:2:r:AI2
- R:752:rw:Control
- R_b0:256:r:Ready
- R_b1:256:r:Run
- R_b2:256:r:Trip
- R_b3:256:r:Warning
- R:282:r>Error1
- R:283:r>Error2
- R:284:r>Error3
- R:1001:r:LinearV

Рис. 3.6 Приклад реалізації модулю збору інформації ПЗ OpenSCADA на сервері

Як видно на зображенні, сервер зчитує вже оптимізовану інформацію з плюзу даних ПЗ OpenSCADA встановленого на Raspberry Pi, цим самим зменшується обсяг трафіку, що передається, та навантаження на обчислювальну потужність серверу, що має обраховувати безліч різних об'єктів.

Наступним етапом роботи сервера є запис отриманої інформації в базу даних, за що відповідають відповідні функції відправки запитів.

```
using Special.FLibSYS;

status = ModBus.Lepohovka7skv.ch1.DI1 && ModBus.Lepohovka7skv.ch1.DI2 && ModBus.Lepohovka7skv.ch1.DI3 && ModBus.Lepohovka7skv.ch1.DI4 && ModBus.Lepohovka7skv.ch1.DI5 && ModBus.Lepohovka7skv.ch1.DI6 && ModBus.Lepohovka7skv.ch1.DI7
&& ModBus.Lepohovka7skv.ch1.DI8 && ModBus.Lepohovka7skv.ch1.DI9 && ModBus.Lepohovka7skv.ch1.DI10 && ModBus.Lepohovka7skv.ch1.DI11 && ModBus.Lepohovka7skv.ch1.DI12 && ModBus.Lepohovka7skv.ch1.DI13 && ModBus.Lepohovka7skv.ch1.DI14
&& ModBus.Lepohovka7skv.ch1.DI15 && ModBus.Lepohovka7skv.ch1.DI16;

rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Ready, 0, "skv7leph", "Rb025e", "phistory_skv7leph");
rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Run, 0, "skv7leph", "Rb125e", "phistory_skv7leph");
rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Trip, 0, "skv7leph", "Rb225e", "phistory_skv7leph");
rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Warning, 0, "skv7leph", "Rb325e", "phistory_skv7leph");
if (ModBus.Lepohovka7skv.ch1.Current1 == 1 || ModBus.Lepohovka7skv.ch1.Current2 == 1 || ModBus.Lepohovka7skv.ch1.LinearV == 1 || status == 1)
{
    rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(100, 0, "skv7leph", "ConnectionWithPP", "phistory_skv7leph");
}
else
{
    rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Current1, 0, "skv7leph", "R100", "phistory_skv7leph");
    rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Current2, 0, "skv7leph", "R101", "phistory_skv7leph");
    rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.LinearV, 0, "skv7leph", "R1001", "phistory_skv7leph");
    rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Error1, 0, "skv7leph", "R202", "phistory_skv7leph");
    rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Error2, 0, "skv7leph", "R203", "phistory_skv7leph");
    rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(ModBus.Lepohovka7skv.ch1.Error3, 0, "skv7leph", "R204", "phistory_skv7leph");
    if (ModBus.Lepohovka7skv.ch1.Current1 == -1.79E307 || ModBus.Lepohovka7skv.ch1.Current1 == -1.79E308 || ModBus.Lepohovka7skv.ch1.Current1 == -1.79E97)
    {
        rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(100, 0, "skv7leph", "ConnectionWithPP", "phistory_skv7leph");
    }
    else
    {
        rez=DQJ.AvalikeCalc.lib_func.upd_arhiv_float_skv(0, 0, "skv7leph", "ConnectionWithPP", "phistory_skv7leph");
    }
}
}
```

Рис. 3.7 Приклад функції для запису інформації в таблицю історії

```
Програма:
using Special.FLibSYS;
//Получим системные дату и время
date_time=syscall("date +%Y-%m-%d %H:%M:%S");
if (date_time!="0000-00-00 00:00:00")
{
    f_in=real2str(f_in,kol_zn,"f");
    if (f_in != -1.79E308 && f_in != -1.79E97 && f_in != -1.79E307)
    {
        //Определить ID объекта
        rez=dbReqSQL("MySQL.MySQL","Select ID,CLASS_ID From objects where TITLE='"+f_name+"';",trans=EVAL);
        f_id_object=rez[1][0];
        f_class_id=rez[1][1];

        //Определить ID свойства объекта
        rez=dbReqSQL("MySQL.MySQL","Select ID From properties where TITLE='"+f_param+"' AND CLASS_ID='"+f_class_id+"';",trans=EVAL);
        f_id_properties=rez[1][0];

        //Определить ID таблицы pvalues
        rez=dbReqSQL("MySQL.MySQL","Select ID From pvalues where OBJECT_ID='"+f_id_object+"' and PROPERTY_ID='"+f_id_properties+"';",trans=EVAL);
        f_id=rez[1][0];

        rez=dbReqSQL("MySQL.MySQL","Select VALUE From pvalues where ID='"+f_id+"';",trans=EVAL);
        answer=rez[1][0];
        if (str2real(answer)!=f_in){
            zapros="objects/?object='"+f_name+"&op=set&p='"+f_param+"&v='"+f_in;
            req = SYS.XMLNode("GET");
            req.setAttr("URI",zapros);
            SYS.Transport.Sockets.out_Server_out3.messIO(req,"HTTP");
            rez=dbReqSQL("MySQL.MySQL","INSERT INTO "+f_table+" (VALUE_ID, ADDED, VALUE) VALUES('"+f_id+"','"+date_time+"','"+f_in+"');",trans=EVAL);
        }
    }
}
```

Рис. 3.8 Шаблон функції для роботи з базою даних

Дві створених функції реалізують алгоритми запису поточних значень змінних в дві таблиці: таблицю поточних даних та таблицю історичних даних. Одночасне виконання двох функцій для кожного об'єкту забирає більше обчислювальної потужності сервера, але зменшує кількість необхідних операцій в майбутньому:

- Таблиця поточних даних містить інформацію про всі змінні системи, їх значення та останній час оновлення, що значно пришвидшує процес зчитування інформації для коректного і своєчасного її відображення в графічному інтерфейсі користувача.
- Ведення таблиці історичних даних зменшує час та кількість запитів до бази даних, що необхідна для створення графіків чи трендів.

ID	PROPERTY_ID	OBJECT_ID	VALUE	UPDATED	PROPERTY_NAME	LINKED_MODULES	SOURCE
<input type="checkbox"/>	8204	8204	82	0	2019-10-21 15:43:36	skv9novobob.Rb3256	
<input type="checkbox"/>	8203	8203	82	0	2019-10-21 15:43:32	skv9novobob.Rb2256	
<input type="checkbox"/>	8202	8202	82	0	2019-11-21 13:22:12	skv9novobob.Rb1256	/objects/object+skv9novobob...
<input type="checkbox"/>	8201	8201	82	1	2019-10-21 18:26:04	skv9novobob.Rb0256	
<input type="checkbox"/>	8111	8111	81	0	2019-11-21 21:49:11	skv11prysh.ConnectionWithPP	/objects/object+skv11prysh...
<input type="checkbox"/>	8110	8110	81	426	2019-11-21 21:49:11	skv11prysh.R1001	
<input type="checkbox"/>	8109	8109	81	20	2019-10-21 18:26:06	skv11prysh.R284	/objects/object+skv11prysh...
<input type="checkbox"/>	8108	8108	81	20	2019-10-21 18:26:05	skv11prysh.R283	
<input type="checkbox"/>	8107	8107	81	20	2019-10-21 18:26:05	skv11prysh.R282	/objects/object+skv11prysh...
<input type="checkbox"/>	8106	8106	81	0	2019-11-21 21:49:10	skv11prysh.R101	
<input type="checkbox"/>	8105	8105	81	0	2019-11-21 21:49:10	skv11prysh.R100	/objects/object+skv11prysh...
<input type="checkbox"/>	8104	8104	81	0	2019-10-21 15:42:35	skv11prysh.Rb3256	
<input type="checkbox"/>	8103	8103	81	0	2019-10-21 15:42:29	skv11prysh.Rb2256	
<input type="checkbox"/>	8102	8102	81	0	2019-10-23 16:13:30	skv11prysh.Rb1256	
<input type="checkbox"/>	8101	8101	81	1	2019-10-21 18:26:05	skv11prysh.Rb0256	/objects/object+skv7lepeh...
<input type="checkbox"/>	8011	8011	80	100	2019-11-21 21:49:22	skv7lepeh.ConnectionWithPP	/objects/object+skv7lepeh...
<input type="checkbox"/>	8010	8010	80	404	2019-11-21 14:14:22	skv7lepeh.R1001	
<input type="checkbox"/>	8009	8009	80	17	2019-10-21 17:43:12	skv7lepeh.R284	/objects/object+skv7lepeh...
<input type="checkbox"/>	8008	8008	80	17	2019-10-21 18:26:02	skv7lepeh.R283	
<input type="checkbox"/>	8007	8007	80	17	2019-10-21 18:26:02	skv7lepeh.R282	/objects/object+skv7lepeh...
<input type="checkbox"/>	8006	8006	80	0	2019-11-21 14:14:22	skv7lepeh.R101	
<input type="checkbox"/>	8005	8005	80	0	2019-11-21 14:14:22	skv7lepeh.R100	/objects/object+skv7lepeh...
<input type="checkbox"/>	8004	8004	80	0	2019-10-21 15:43:23	skv7lepeh.Rb3256	
<input type="checkbox"/>	8003	8003	80	0	2019-10-21 16:16:30	skv7lepeh.Rb2256	
<input type="checkbox"/>	8002	8002	80	0	2019-11-19 04:06:56	skv7lepeh.Rb1256	/objects/object+skv7lepeh...

Выбрать все Инвертировать выбор

Results generated by SQL query: `select * from 'pvalues' order by 'ID' desc limit 280,25`

Edit selected row Add row Delete selected row

Поиск строк, где поле ID содержит

Advanced Search

Перейти на строку:

Вернуться к списку полей Вернуться к списку таблиц Вернуться к списку баз данных

Рис. 3.9 Вигляд таблиці поточних значень в БД

ID	VALUE_ID	ADDED	VALUE
<input type="checkbox"/>	1	51733	2019-11-07 13:26:10
<input type="checkbox"/>	2	51732	2019-11-07 13:26:11
<input type="checkbox"/>	3	51733	2019-11-07 13:27:10
<input type="checkbox"/>	4	51734	2019-11-07 13:28:27
<input type="checkbox"/>	5	51735	2019-11-07 13:28:35
<input type="checkbox"/>	6	51736	2019-11-07 13:28:41
<input type="checkbox"/>	7	51733	2019-11-07 13:29:15
<input type="checkbox"/>	8	51732	2019-11-07 13:29:16
<input type="checkbox"/>	9	51732	2019-11-07 13:31:10
<input type="checkbox"/>	10	51732	2019-11-07 13:32:10
<input type="checkbox"/>	11	51732	2019-11-07 13:33:10
<input type="checkbox"/>	12	51733	2019-11-07 13:34:10
<input type="checkbox"/>	13	51742	2019-11-07 13:41:53
<input type="checkbox"/>	14	51743	2019-11-07 13:43:15
<input type="checkbox"/>	15	10408	2019-11-07 13:44:11
<input type="checkbox"/>	16	10408	2019-11-07 13:45:10
<input type="checkbox"/>	17	10409	2019-11-07 13:46:10
<input type="checkbox"/>	18	51744	2019-11-07 13:47:05
<input type="checkbox"/>	19	51745	2019-11-07 13:47:14
<input type="checkbox"/>	20	10409	2019-11-07 13:47:15
<input type="checkbox"/>	21	10408	2019-11-07 13:47:16
<input type="checkbox"/>	22	51746	2019-11-07 13:47:18
<input type="checkbox"/>	23	10409	2019-11-07 13:48:10
<input type="checkbox"/>	24	10409	2019-11-07 13:49:10
<input type="checkbox"/>	25	10408	2019-11-07 13:49:11

Выбрать все Инвертировать выбор

Results generated by SQL query: `select * from 'phistory_pns_inter' limit 0,25`

Edit selected row Add row Delete selected row

Поиск строк, где поле ID содержит

Advanced Search

Вернуться к списку полей Вернуться к списку таблиц Вернуться к списку баз данных

Рис. 3.10 Вигляд таблиці історичних значень в БД

Наступним етапом є передача записаної в БД інформації до веб-сервера. Веб-сервер, що працює на сервері за допомогою запитів, що написані на мові PHP, зчитує з бази даних необхідну інформацію, та шляхом HTTP запитів надсилає їх до графічного інтерфейсу користувача.

Для оптимізації процесу створення графіків і трендів на мові PHP було реалізовано створення JSON структури. Дане перетворення зменшує кількість даних, що передаються, та істотно зменшує час формування графіків.

```
//Получаем начальную дату
$start_date=gg("skv9novoob.grafik_data_start");
//Получаем конечную дату
$end_date=gg("skv9novoob.grafik_data_end");

// Получить имя и проверить
//isset — Определяет, была ли установлена переменная значением,
отличным от NULL https://www.php.net/manual/ru/function.isset.php
if ( isset($params['name']) ) { $name = $params['name']; } else { return;
}

// Разбить на объект и свойство
//explode — Разбивает строку с помощью разделителя
https://www.php.net/manual/ru/function.explode.php
$name = explode('.', $name);

// Получить объект по имени
$obj=getObject($name[0]);
// Получить id свойства
$prop_id=$obj->getPropertyByName($name[1], $obj->class_id, $obj-
>id);

// Получаем VALUE_ID для таблицы истории
$pvalue=SQLSelectOne("SELECT * FROM pvalues WHERE
PROPERTY_ID='".$prop_id.'" AND OBJECT_ID='".$obj->id.'";");
// Получаем данные из таблицы истории
$arr_s = SQLSelect("SELECT UNIX_TIMESTAMP(ADDED) as
ADDED, VALUE FROM phistory_skv9novoob WHERE
VALUE_ID='".$pvalue['ID'].'" AND ADDED>=('.$start_date." 00:00:00')
AND ADDED<=('.$end_date." 23:59:59') ORDER BY ADDED");
```

```

// Собрать структуру JSON
$st = $params['callback'].('[';
$f = false;
//https://www.php.net/manual/ru/control-structures.foreach.php
foreach($arr_s as $s) {
    if ($f) { $st .= ','; } else { $f = true; }
    $st .= '['.$s['ADDED']. '000,'.$s['VALUE'].']' ;
}
$st .= ']);

// Ответ
echo ($st);

```

Даний алгоритм створення структур дозволяє суттєво зменшити кількість трафіку, пришвидшити швидкість формування графіків в графічному інтерфейсі користувача та пришвидшити роботу системи графічного відображення в цілому, оскільки при виборі іншого алгоритму реалізації, данні зберігаються в кеші, та помітно пригальмовують роботу системи.

Далі приведено приклад програми для обробки стану свердловини, що керується за допомогою пристрою плавного пуску Schneider Electric Altistart 22.

```

$name = $this->object_title;
$obj = getObject($name);
$a = $obj->GetProperty('Rb0256');
$b = $obj->GetProperty('Rb1256');
$c = $obj->GetProperty('Rb2256');
$d = $obj->GetProperty('R282');

if ($c == 1 && $d == 1)
{
    $obj->SetProperty('Status', 'Аварія');
    $obj->SetProperty('AlarmDiscription', 'Недонавантаження по
струму');
}
if ($c == 1 && $d == 2)
{
    $obj->SetProperty('Status', 'Аварія');

```

```

        $obj->SetProperty('AlarmDiscription',      'Перенавантаження      по
струму');
    }
    if ($c == 1 && $d == 3)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Асиметрія фаз');
    }
    if ($c == 1 && $d == 4)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Відключення      по      струму
витоку');
    }
    if ($c == 1 && $d == 5)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Перенавантаження двигуна');
    }
    if ($c == 1 && $d == 6)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Перегрів двигуна');
    }
    if ($c == 1 && $d == 7)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Перегрів двигуна (РТС));
    }
    if ($c == 1 && $d == 8)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Чергування фаз');
    }
    if ($c == 1 && $d == 9)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Втрата фази');
    }

```



```

    }
    if ($c == 1 && $d == 10)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Низька напруга чи її
відсутність');
    }
    if ($c == 1 && $d == 11)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Висока напруга');
    }
    if ($c == 1 && $d == 12)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Невідповідність часу запуску');
    }
    if ($c == 1 && $d == 13)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Завелика кількість пусків');
    }
    if ($c == 1 && $d == 14)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Замикання теристорів або
неправильне підключення');
    }
    if ($c == 1 && $d == 15)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Зовнішня помилка');
    }
    if ($c == 1 && $d == 16)
    {
        $obj->SetProperty('Status', 'Аварія');
        $obj->SetProperty('AlarmDiscription', 'Внутрішня помилка');
    }

```

```

        if ($c == 1 && $d == 17)
        {
            $obj->SetProperty('Status', 'Аварія');
            $obj->SetProperty('AlarmDiscription', 'Перевищений час передачі по
ModBus');
        }
        if ($c == 1 && $d == 18)
        {
            $obj->SetProperty('Status', 'Аварія');
            $obj->SetProperty('AlarmDiscription', 'Код переривання');
        }
        if ($c == 1 && $d == 19)
        {
            $obj->SetProperty('Status', 'Аварія');
            $obj->SetProperty('AlarmDiscription', 'Коротке замикання');
        }
        if ($c == 1 && $d == 20)
        {
            $obj->SetProperty('Status', 'Аварія');
            $obj->SetProperty('AlarmDiscription', 'Несправність байпасного
контрактора');
        }
        if ($c == 1 && $d == 21)
        {
            $obj->SetProperty('Status', 'Аварія');
            $obj->SetProperty('AlarmDiscription', 'Некоректна конфігурація');
        }
    }

```

Як видно з алгоритму обробки, базуючись на отриманих з БД даних в результаті роботи програми на виході веб-клієнт отримує зрозуміле представлення про стан даного об'єкту, що зменшує кількість інформації, що передається, мінімізує навантаження на обчислювальну потужність комп'ютера користувача, та дозволяє системі реального часу своєчасно реагувати на зміну параметрів системи.

3.3 Опис верхнього рівня автоматизації

Верхній рівень автоматизації, в тому числі алгоритми аналізу та графічного відображення інформації реалізовані на програмного забезпеченні MajorDoMo (Major Domestic Module – головний внутрішній модуль). Дане ПЗ є середовищем розробки людино-машинних інтерфейсів, що представляє собою набір інструментів, що допомагають в структуруванні і створенні алгоритмів обробки інформації, створенні сцен відображення, реалізації реакцій на певні події, та створення моделей поведінки в певних ситуаціях.

Доступ до системи диспетчеризації можна отримати за допомогою будь-якого браузера з комп'ютера чи іншого пристрою, що підключений до мережі VPN, що значно спрощує і розширює можливості експлуатації системи.

Після введення адреси в пошукову строку браузер відкриває головну сторінку системи диспетчеризації.

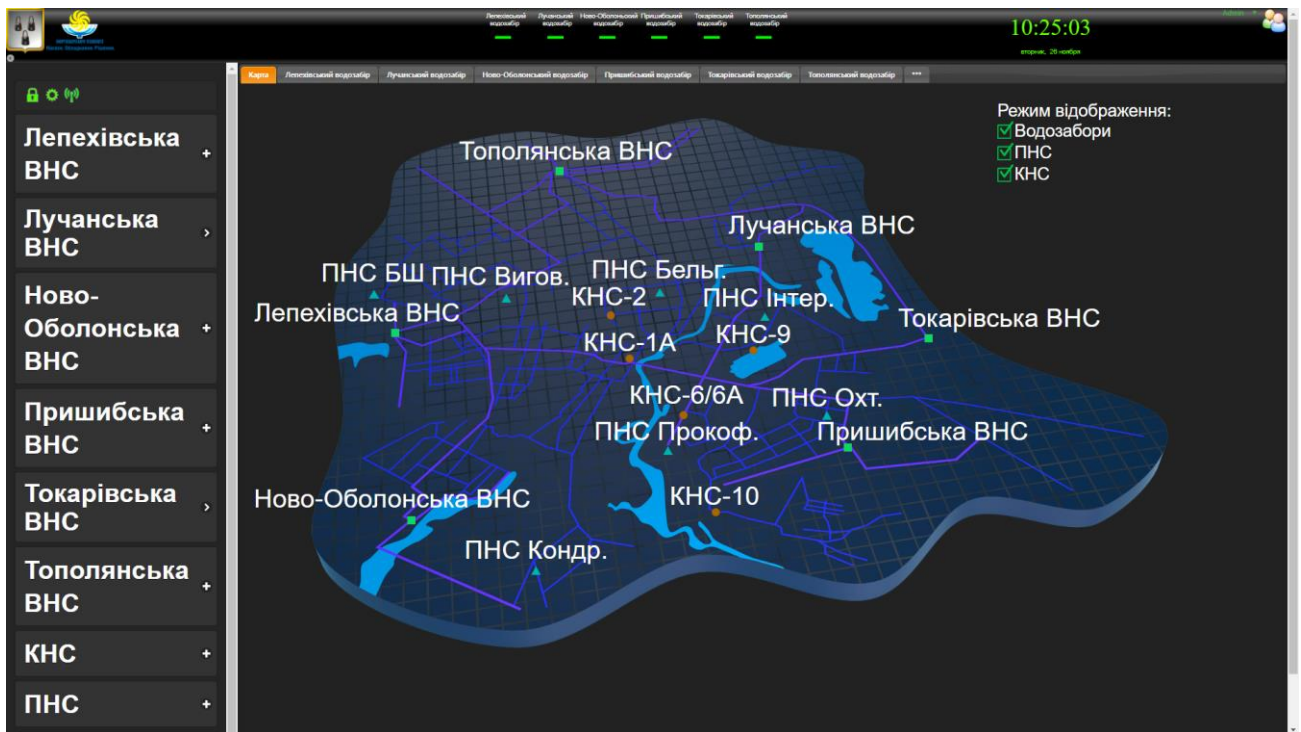


Рис 3.11 Головний екран системи диспетчеризації

На головній сторінці відображаються всі об'єкти диспетчеризації, їх розташування на карті та їх стан (білий колір всіх написів свідчить про відсутність аварійних станів).

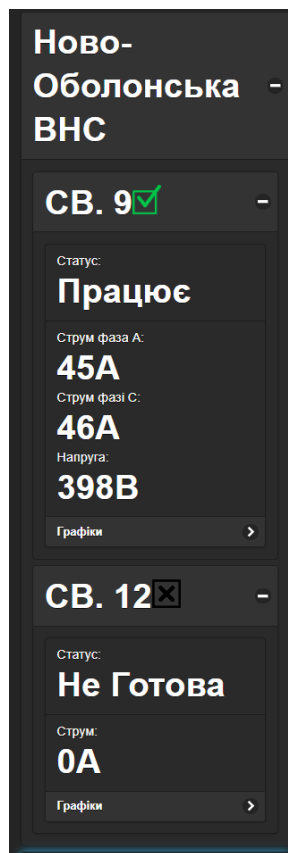


Рис. 3.12 Бокова панель системи диспетчеризації

В лівій частині екрану знаходиться панель швидкого доступу, що дозволяє оперативно переглядати поточний стан об'єктів, їх параметри об'єктів та графіки. Дана функція реалізована для зручності перегляду станів всіх свердловин і інших об'єктів у вигляді списку. Також дана функція дозволяє одночасно переглядати стан свердловин на різних водозаборах, адже в головному інтерфейсі стан свердловини можна переглянути лише відкривши вікно її водозабору.

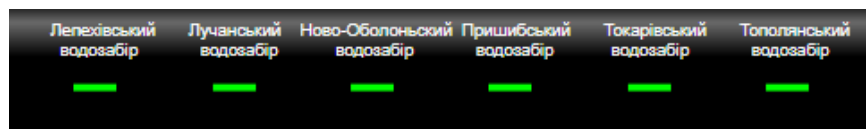


Рис. 3.13 Інформаційне табло стану водозаборів

В верхній частині екрану завжди знаходяться індикатори роботи водозаборів, що сигналізують про стан, в якому наразі знаходяться водозабори. При виникненні аварійних ситуацій на каналізаційних насосних станціях, що можуть спричинити екологічну катастрофу, в системі диспетчеризації з'являється спливаюче вікно, з поясненням місця і причини виникнення аварії, що супроводжується звуковими сигналами для привернення уваги.

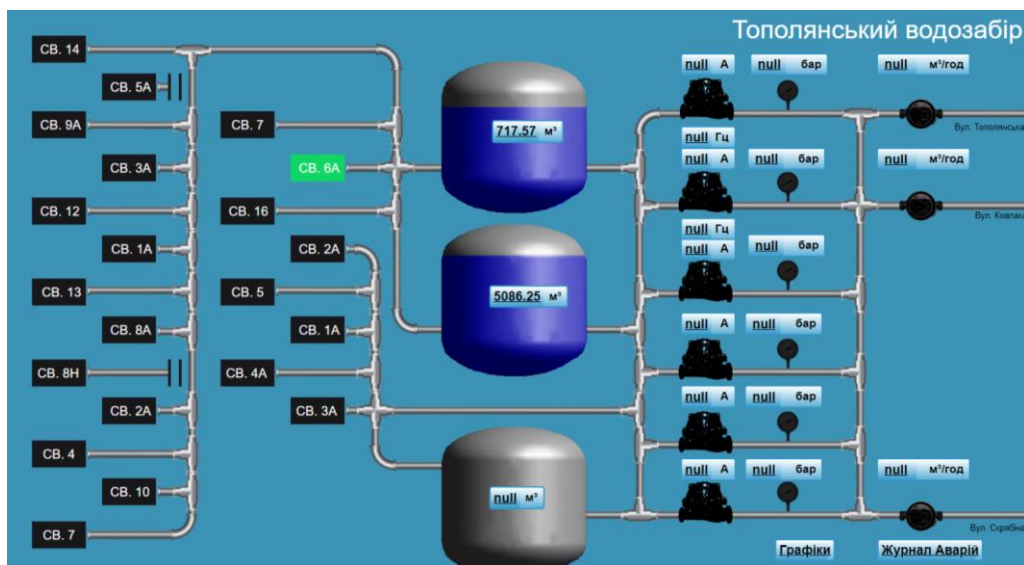


Рис. 3.14 Функціональна схема водозабору

На даному зображенні приведена функціональна схема водозабору, на якій відображаються стан роботи свердловин, рівень води в резервуарах чистої води, витрата на вихідних трубопроводах, тиск на виході з насосів, а також стан і струм на насосах, що працюють на прямих пусках, і додатково відображення частоти напруги живлення для насосів, що працюють на частотних перетворювачах.

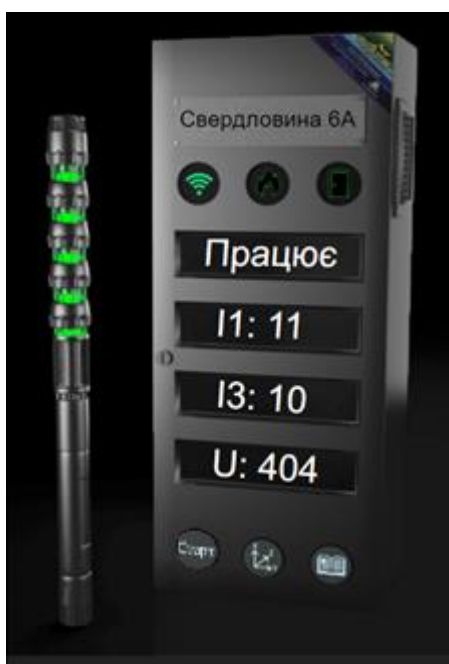


Рис. 3.15 Вікно свердловини

Для переходу у вікно свердловини необхідна натиснути на відповідну іконку в вікні водозабору. При цьому з'являється анімоване зображення свердловинного насосу, що показує поточний стан роботи свердловини, та шафа відображення даних, що відображає стан свердловини, стан зв'язку з свердловиною, а також струм і напругу, що наразі надходить на двигун насосу. Крім цього у нижній частині шафи знаходяться кнопки дистанційного керування свердловиною, переходу у журнал аварій і перегляду графіків.

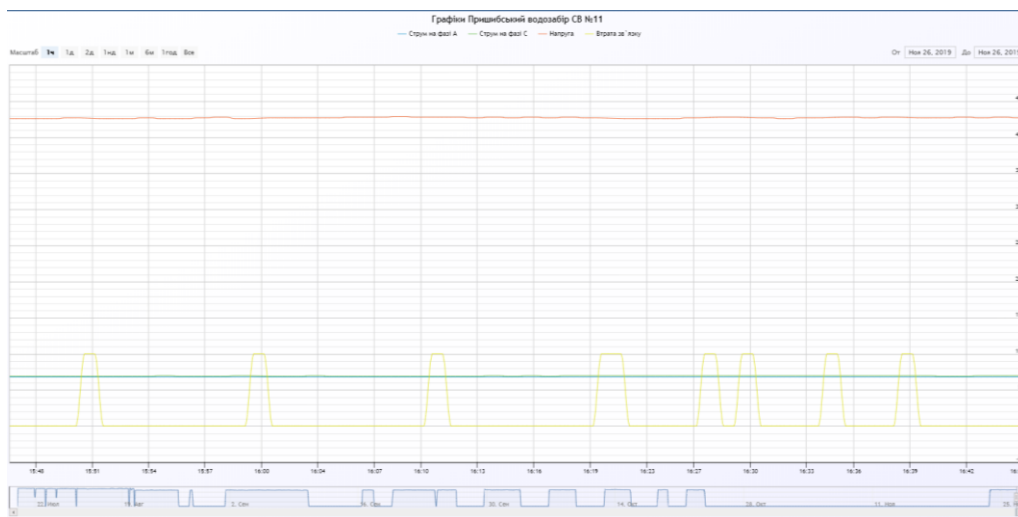


Рис. 3.16 Вікно графіків свердловини

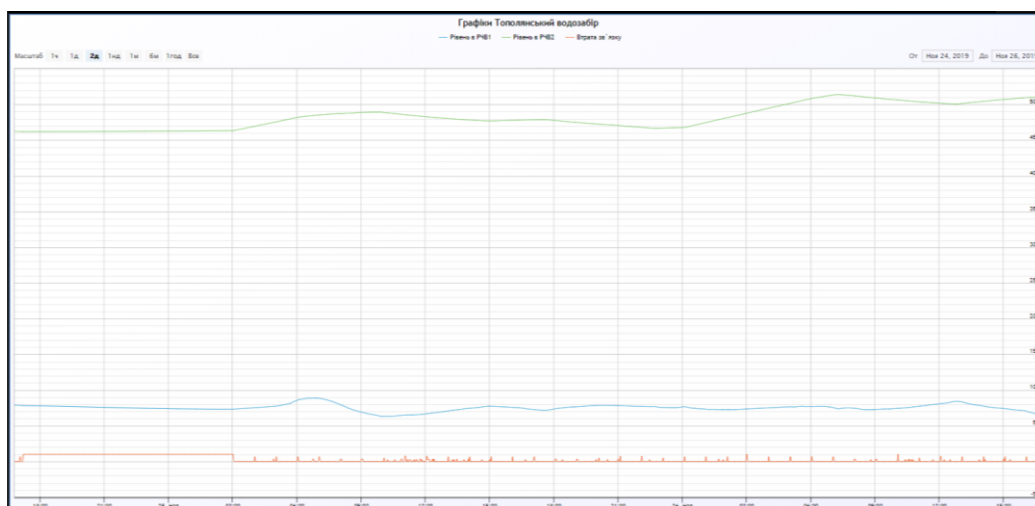


Рис. 3.17 Вікно графіків водозаборів

На даних графіках можна відслідкувати зміну параметрів в реальному часі, та аналізувати історичну зміну параметрів та відслідковувати поведінку об'єкта до появи на ньому аварії чи несправності.

H	1h	24h	7d	31d
Оптимізувати лог				
T		V		
2019-11-01 17:41:08		Некоректна конфігурація		
Load all values				

Рис. 3.18 Журнал аварій

Ведення журналу аварій дозволяє в зручному вигляді відображати всі несправності, що виникали на певному об'єкті, що дозволяє аналізуючи всі несправності на свердловині приймати рішення про необхідність заміни силового чи виконавчого обладнання.



Рис. 3.19 Вікно ПНС

В даному вікні відображається поточний стан і інформація отримані з ПНС, в тому числі стан ПНС, стан насосів, що працюють в резервно-піковому режимі, та тиск води в вихідному трубопроводі, створення якого необхідно для забезпечення водопостачання у багатоповерхових будинках.



Рис. 3.20 Вікно КНС

В даному вікні відображається стан каналізаційної насосної станції та стан роботи насосів. Крім цього реалізовано відображення рівня в грабельній, напруги на виході з трансформатора, та струм на фазах третього двигуна.

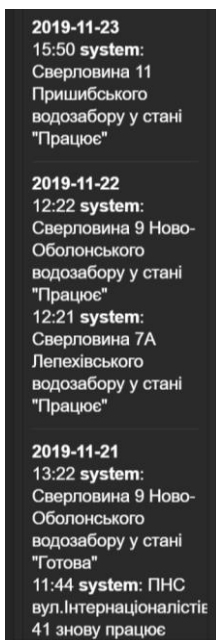


Рис. 3.19 Журнал подій

Окрім журналу аварій, був реалізований журнал подій, в якому збирається інформація щодо всіх змін, що сталися на всіх об'єктах. Це дозволяє оцінювати загальний стан всіх об'єктів, та отримувати необхідну інформацію для прийняття рішення щодо здійснення керуючого впливу на інші об'єкти. Крім цього, ведення журналу дозволяє відслідковувати послідовність подій, що призвела до виникнення надзвичайних ситуацій.

4 РОЗРОБКА СТАРТАП-ПРОЕКТУ ПО АВТОМАТИЗАЦІЇ СИСТЕМИ ВОДОПОСТАЧАННЯ І ВОДОВІДВЕДЕННЯ МІСТА

4.1 Основні положення ідеї

Основним завданням створення даного проекту є створення універсальної системи автоматизації, що може бути впроваджена на будь-якому обладнанні, без необхідності його заміни. Створена система реалізує алгоритми збору інформації, аналізу та дистанційного керування об'єктами автоматизації. Для реалізації ідеї необхідним є лише придбання серверу, та закупівля або шаф збору інформації, що включають в себе одноплатний комп'ютер та модулі вводів/виводів, або шаф керування, що додатково мають контролер. Топологія промислової мережі робить не обов'язковим прокладання провідних сполучень і дозволяє підключати до системи віддалені об'єкти, що є необхідною умовою для створення розподіленої системи автоматизації.

4.2 Опис області застосування проекту

Даний проект може бути реалізований на будь-якій системі водопостачання і водовідведення (село, селище, місто, громада, область, країна). Крім того, дана архітектура і програмне забезпечення може бути використане для створення системи автоматизації будь-якого іншого підприємства, що потребує реалізації розподіленої системи автоматизації.

4.3 Основні переваги для замовників

Основними перевагами для замовників даного проекту є реалізація можливості дистанційного збору інформації та дистанційного керування об'єктами автоматизації, що як наслідок веде до можливості зменшення обслуговуючого персоналу на об'єктах, та можливості оперативного реагування

на аварійні обставини, що дозволяє зменшити час на їх виправлення, та як наслідок останніх двох пунктів, економити кошти та мінімізувати можливість людської помилки. Крім того, перевагою для замовників, що оберуть цей проект, є те, що даний проект може бути реалізований на вже встановленому обладнанні, що зменшує його собівартість і терміни окупності.

4.4 Аналіз конкурентів та порівняння системи з існуючими

Для проведення аналізу буде створено порівняльну таблицю, в якій ведеться порівняння сильних і слабких техніко-економічних показників різних конкурентів, що представлені на даному ринку збуту.

В результаті порівняння характеристик буде зроблено рішення щодо конкурентоспроможності різних сторін даного проекту, що будуть позначені в таблиці як:

Н – недолік проекту в порівнянні з конкурентами;

Е – еквівалент рішень, що представлені конкурентами;

П – перевага даного проекту над іншими.

Таблиця 4.1 Визначення конкурентоспроможності проекту

№	Техніко-економічні характеристики	Пропозиції конкурентів			Висновок
		Цей проект	Конкурент 1	Конкурент 2	
1.	Собівартість	Низька	Висока	Висока	П
2.	Результат виконання	Веб-сторінка	Програма для НМІ-панелі	Програма для ПК	Е
3.	Централізована система	Так	Ні	Так	П

4.	Наявність обслуговуючого персоналу на об'єктах	Не потрібна	Потрібна	Не потрібна	П
5.	Надійність системи	Середня	Висока	Середня	С
6.	Легкість впровадження	Висока	Середня	Середня	П
6.	Необхідність заміни існуючого обладнання	Не потрібна	Потрібна	Потрібна	П
7.	Швидкість реалізації	Низька	Висока	Середня	С

Як видно з таблиці, сильними сторонами даного проекту є відсутність необхідності зміни обладнання, що як наслідок веде до меншої собівартості кінцевого продукту. Крім того, за рахунок відмови від локального обслуговуючого персоналу і централізації системи на станції оператора, система дозволяє скоротити штат і налагодити роботу системи в цілому. Недоліками системи є низька швидкість його реалізації, що є наслідком відмови від зміни обладнання, оскільки доводиться працювати з обладнанням різних виробників. Також через передачу інформації за допомогою мобільного зв'язку, система вразлива від якості зв'язку, що може спричинити тимчасову втрату зв'язку з об'єктом, тому вона значно поступається в надійності конкурентам, що використовують провідне з'єднання, але виграє в них по легкості реалізації підключення.

4.5 Технологічний аудит ідеї проекту

В даному підрозділі буде проведено аудит технологій, що необхідні для реалізації проекту, та дано відповіді на питання, щодо доступності технологій, їх довершеності, та наявності готових рішень для спрощення реалізації кінцевого продукту.

Таблиця 4.2 – Можливість технічної реалізації проекту

№	Частина проекту	Необхідні технології	Стан технології
1	Розробка програмного забезпечення для контролера	CoDeSys V2.3	Безкоштовна програма у відкритому доступі
2	Створення операційного забезпечення для Raspberry Pi	OpenWRT	Безкоштовний SDK у відкритому доступі
3	Створення програмного забезпечення для Raspberry Pi	OpenSCADA	Безкоштовна програма у відкритому доступі
4	Створення операційного забезпечення для серверу	OS Linux	Безкоштовна ОС у відкритому доступі
5	Створення програмного забезпечення для серверу	OpenSCADA	Безкоштовна програма у відкритому доступі
6	Створення бази даних	MariaDB	Безкоштовна БД у відкритому доступі
7	Створення VPN мережі	PPTP	Безкоштовна програма у відкритому доступі
8	Створення HMI інтерфейсу	MajorDoMo	Безкоштовна програма у відкритому доступі

Як видно, всі використаними технологіями можна користуватись безкоштовно, що зменшує кількість коштів, що необхідні для реалізації проекту, а також його подальшого використання. Крім того, всі використані програми мають перевірений та довершений функціонал, що дозволяє реалізовувати всі необхідні складові, що необхідні для створення проекту.

4.6 Аналіз ринкових можливостей для запуску стартап-проекту

Даний аналіз допоможе виокремити бажаний сегмент ринку продажу та допомогти знайти потенціальних замовників. Передчасний аналіз ринкових можливостей дозволяє запланувати стратегію розвитку проекту, проаналізувати його сильні сторони, на яких необхідно зробити акцент, та виокремити слабкі сторони, що роблять його менш конкурентоспроможним.

Для початку виокремимо потенціальних замовників проекту.

Таблиця 4.3 – Характеристика потенційних замовників

№	Потреба	Цільова аудиторія	Особливість у поведінці цільових груп	Вимоги до кінцевого продукту
1	Програмне забезпечення, що дозволяє збирати	Місцеві Водоканали	Проект реалізується для забезпечення технологічного процесу	Висока надійність та великий функціонал
2	інформацію та керувати розподіленою системою автоматизації	Приватні замовники	Проект реалізується для моніторингу віддаленого об'єкту автоматизації (будинок, дача)	Можливість перегляду в мобільному додатку

Другим пунктом аналізу ринку є оцінка попиту на дану продукцію, та обсяг ринку.

Таблиця 4.4 – Аналіз ринку збуту

№	Показники ринку збуту	Характеристика
1	Кількість великих конкурентів, од	4
2	Кількість великих потенціальних покупців, од	~500
3	Динаміка ринку (якісна оцінка)	Зменшується
4	Максимальна кількість проектів/ рік	3-4
5	Середня рентабельність в галузі	25%

Як видно з таблиці, наразі існує не велика кількість потенціальних замовників даного проекту, але через невелику конкуренцію і відносно великі терміни, що необхідні для реалізації проекту, наразі попит на ринку значно перевищує пропозицію. А зважаючи на старіння технологій, та підвищення рівня вимог до систем автоматизації, замовники на ринку будуть присутні завжди.

Таблиця 4.5 – Фактори загроз

№	Фактор	Загроза	Можливе вирішення
1	Конкуренція	Втрата клієнтів	Створити більш конкуренто-спроможну пропозицію
2	Старіння технологій	Втрата інтересу замовників до запропоновано продукту	Безперервне вдосконалення програмно-технічних рішень для створення конкуренції
3	Втрата постачальника	Неможливість отримати пристрої/ПЗ для реалізації проекту	Постійний моніторинг аналогів пристроїв/ПЗ, що використовуються

Таблиця 4.6 – Фактори можливостей

№	Фактор	Можливість	Можлива вигода
1	Розширення зони діяльності	Вихід на міжнародний ринок	Значне збільшення кількості потенційних замовників
2	Монополізація регіону	Запропонувати рішення, яке найбільше підходить для певного регіону	Відсутність конкуренції
3	Альтернативність	Запропонувати інше, більш дешеве/дороге вирішення поставленої задачі проекту	Збільшення кола зацікавлених замовників

Для внесення ясності на актуальність можливості виходу на ринок, необхідним є аналіз конкуренції, що вже є на цьому ринку.

Таблиця 4.7 – Ступеневий аналіз конкуренції

№	Особливості конкуренції ринку	Фактор прояву	Вплив факторів на стратегію підприємства
1	Тип конкуренції - досконалий	Існує 2 великих компанії-конкуренти	Необхідність корегування цінової політики та пошук відповідної ніші на ринку
2	За рівнем боротьби - міжнародний	Всі компанії-конкуренти є іноземними	Велика можливість зайняти місце на локальному ринку збуту

Продовження таблиці 4.7			
3	За галузевою ознакою - внутрішня	Наявність готових рішень у конкурентів в даній галузі	Універсалізація програмно-технічного комплексу, для його використання в інших галузях
4	Вид товарів – товарно-видовий	Всі запропоновані рішення є ідентичними в кінцевому результаті (відмінність лише в реалізації)	Доведення процесу створення і кінцевого продукту до найвищої якості, для створення переваги над конкурентами
5	Характер переваг - неціновий	В реалізації даних проєктів більша роль відводиться надійності, ніж витратам	Запропонувати конкурентоздатний варіант реалізації за меншу вартість
6	За інтенсивність - марочна	Фірми конкуренти мають більш відомі імена	Зробити рекламу компанія та заслужити довіру на ринку

Таблиця 4.8 – Аналіз конкуренції за М. Портером

Прямі конкуренти	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Існує два конкуренти, що використовують для реалізації свої пропрієтарні рішення	Існує декілька компаній, що займається схожою діяльністю в інших галузях	Наразі існує безліч поставників необхідних товарів та альтернативних товарів	Наразі існує велика кількість зацікавлених замовників	Наразі всі альтернативні рішення є набагато дорожчими і складнішими в реалізації

Після проведення аналізу, можна зробити висновки, що даний проект є конкурентоспроможним, оскільки є набагато економічно доцільнішим, попри однакові з конкурентами техніко-економічні показники, та показники якості і надійності. Слабкою стороною проекту є відсутність відомого бренду, що зменшує рівень довіри замовника до якості результату.

Таблиця 4.9 – Обґрунтування конкурентоспроможності

№	Фактор	Обґрунтування
1	Можливість реалізації проекту без зміни існуючого обладнання	Наразі в комунальних підприємствах встановлено безліч відносно нового обладнання від різноманітних постачальників. Відмова від зміни вже існуючого обладнання зменшує витрати на створення системи автоматизації в десятки разів

Продовження таблиці 4.9		
2	Безкоштовність програмного забезпечення	Компанії конкуренти беруть щорічну плату за купівлю ліцензії на користування їх програмним забезпеченням, що значно збільшує витрати на експлуатації системи в подальшому
3	Можливість відмови від обслуговуючого персоналу	Система дозволяє відмовитися від локального обслуговуючого персоналу і реалізовувати централізоване керування системою, що зменшує кошти, що витрачаються на виплату заробітних плат
4	Можливість корекції графічного інтерфейсу під окремого замовника	Завдяки середовищу MajorDoMo графічний інтерфейс користувача можна персоналізувати під кожного замовника, що спрощує подальшу роботу з даним ПЗ.

За проведенням вище аналізом зробимо висновки про сильні та слабкі сторони проекту (Таблиця 4.10).

Таблиця 4.10 Висновки про сильні і слабкі сторони проекту

№	Фактор	Бали 1-10	Рейтинг товару конкурентів в порівнянні з даним проектом						
			-3	-2	-1	0	1	2	3
1	Реалізація проекту без зміни обладнання	10							+
2	Безкоштовність ПЗ	6					+		
3	Відмова від обслуговуючого персоналу	5					+		
4	Можливість персоналізації графічного інтерфейсу	4					+		

Продовження таблиці 4.10									
5	Терміни реалізації	2		+					
6	Надійність системи	3		+					

Фінальною частиною аналізу проекту є створення діаграми SWOT-аналізу (Таблиця 4.11), що зручно відображує всі проведені вище аналізи конкурентоспроможності, виокремленню сильних і слабких сторін, аналіз ринку, замовників, можливостей і загроз.

Таблиця 4.11 SWOT – аналіз стартап-проекту

Сильні сторони	Слабкі сторони
<ul style="list-style-type: none"> - Низька вартість реалізації - Немає необхідності заміни обладнання - Мінімізації витрат на обслуговування системи 	<ul style="list-style-type: none"> - Високий термін реалізації проекту - Низька надійність системи
Можливості	Загрози
<ul style="list-style-type: none"> - Монополізація локального ринку - Вихід на міжнародний ринок - Вихід на інші ринки - Створення альтернативної структури системи - Створення більш надійного рішення 	<ul style="list-style-type: none"> - Появ на ринку компанії з альтернативним рішенням

Базуючись на SWOT – аналізі було реалізовано план альтернативної поведінки на ринку для гарантованого виходу проекту на ринок і створення конкуренції існуючим компаніям-представникам.

Таблиця 4.12 Альтернативний план запуску стартап-проекту

№	Коректива	Результат	Ймовірність успіху
1	Введення необхідності на заміну встановленого обладнання	Зменшення терміну реалізації проекту до 6 місяців. Значне збільшення вартості реалізації проекту	30%
2	Необхідність прокладання кабельних сполучень	Збільшення надійності системи. Збільшення терміну підготовчих робіт в півтора рази. Збільшення витрат.	20%
3	Розробка альтернативної архітектури проекту	Можливість надання вибору для замовника. Збільшення можливостей для реалізації. Утримання позицій на ринку	70%

Як видно, третій план альтернативної поведінки подає великі шанси на успіх, тому для закріплення позицій на ринку і створення конкуренції необхідно розпочати створення альтернативного рішення для автоматизації розподіленої системи міського водопостачання і водовідведення, що дасть змогу надати вибір замовникам, та мати вибір у разі появи факторів, що заважають або роблять неможливою реалізацію проекту. Тому створимо базову стратегію розвитку проекту після виходу на ринок.

Таблиця 4.13 – Формування базової стратегії

№	Напрямок розвитку	Стратегія охоплення ринку	Ключові переваги
1	Базовий	Пошук замовників за рахунок більш економічно вигідної пропозиції по створенню системи автоматизації	Наразі в межах України переваги в економічних показниках є досить вагомими, через нестачу фінансування
2	Альтернативний	Надання послуг аналогічних компаніям-конкурентам по якості на більш вигідних умовах	Можливість виходу на міжнародний ринок. Створення ім'я компанії. Надання альтернативного рішення

Після аналізу плану входження на ринок і формування базової стратегії поведінки на ринку, необхідно визначитись зі стратегією позиціонування на ринку (Таблиця 4.14).

Таблиця 4.13 – Стратегія позиціонування

Вимоги до товару	Стратегія розвитку	Ключові переваги проекту	Асоціативний ряд, що має супроводжувати стартап-проект
Надійність, швидкодіяльність	Впровадження альтернативних рішень	Економічна доцільність, гнучкість	Спрощення, централізація, безпечність

Як результат, маємо чітко сформовану стратегію виходу на ринок, що створена на основі аналізу ринку, послуг, що поставляються та компаній-конкурентів. Проаналізовано всі сильні та слабкі сторони, та створено альтернативні рішення, що можуть знівелювати слабкі сторони та загрози для стартап-проекту.

4.7 Розробка маркетингової компанії стартап-проекту

Першим, що необхідно зробити є формування маркетингової концепції для опису товару, який отримує споживач (Таблиця 4.14).

Таблиця 4.14 – Маркетингова концепція

№	Потреба	Вигода, яку отримає замовник	Ключові переваги над конкурентами
1	Можливість дистанційного перегляду параметрів об'єктів системи водопостачання і водовідведення	Можливість аналізу ефективності системи	Можливість реалізації системи без необхідності заміни існуючого обладнання

Продовження таблиці 4.14			
2	Можливість дистанційного керування	Можливість централізованого керування диспетчером	Більш гнучка система
3	Оптимізація роботи системи водопостачання і водовідведення	Можливість аналізу даних по роботі об'єктів з цілю виявлення і покращення ефективності роботи системи	Можливість підключення до системи декількох користувачів, що дозволяє профільним спеціалістам слідкувати за параметрами роботи
4	Швидке реагування на нештатні ситуації	Можливість швидкої реакції на несправності та отримання інформації про природу несправності для більш ефективного реагування	Можливість зчитування аварій з різноманітного обладнання без необхідності його заміни

Для опису і просування товару на ринку було розроблено трирівневу модель опису товару (Таблиця 4.15).

1-й рівень – задум товару, що описує ціль та необхідність введення даної системи на об’єкт, описує проблеми, що дана система може вирішувати, а також приводить аргументи, чому дана система необхідна для впровадження в кожній міській системі водопостачання і водовідведення.

2-й рівень – рівень реального виконання товару, показує кінцевий результат, що буде отриманий після повного закінчення робіт по його введенню.

3-й рівень – супровід товару, рівень що описує можливості, які буде отримувати замовник після реалізації товару.

Таблиця 4.15 – Опис рівнів моделі товару

Рівень товару	Сутність
1.Задум	Реалізація проекту дозволяє збирати дані з об’єктів автоматизації, а також керувати ними. Керування проводиться централізовано, з пункту диспетчерського керування. Система дозволяє збирати і аналізувати отримані данні для оптимізації подальшої роботи підприємства і збільшення економічних показників
2.Реальне виконання	Результатом виконання проекту, буде система автоматизації, що як кінцевий результат буде відображена у якості веб-сторінки в браузері, на якій буде можливість перегляду даних, керування об’єктами та їх аналіз.
3.Супровід	Створений проект матиме довготривалу підтримку зі сторони виконавця. Через можливість дистанційного доступу до серверу можливе оперативне виправлення можливих несправностей, оновлення ПО. Крім того можливе подальше розширення і вдосконалення системи автоматизації по ініціативі замовника.

Після формування маркетингової ідеї слід визначити, яким чином необхідно захистити ідею проекту від можливого копіювання. Захист необхідно реалізувати за рахунок захисту інтелектуальної власності. Новизна даного проекту полягає у введенні оригінального програмно-технічного комплексу для створення подібної системи автоматизації.

Наступник кроком буде опис ринку збуту (Таблиця 4.16). Попереднє вирішення стратегії продажу продукту, конкретизація цільової аудиторії і план по їх пошуку значно спрощують подальші продажі товару та збільшують шанс на успіх стартап-проекту.

Таблиця 4.16 – Опис ринку збуту

Специфіка замовників	Функція збуту, що має виконуватись постачальником товару	Глибина каналу збуту	Оптимальна системи збуту
Замовниками є комунальні підприємства – міські водоканали, що колективно приймають рішення про необхідність впровадження подібної системи	Необхідно проводити показові презентації, виступати на виставках і конференціях для привернення уваги замовників	Збут товару можливий лише напряму, оскільки виконання товару лежить на компанії - виконавці	Оптимальною системою збуту є переконання замовника в необхідності реалізації даного проекту на його підприємстві

Важливим фактором в вибранні стратегії маркетингового просування є вибір шляху комунікації з безпосереднім замовником (Таблиця 4.17).

Специфіка цільових замовників	Канали комунікації	Ключові позиції, обрані для позиціонування	Завдання реклами	Концепція рекламного звернення
Прийняття колективного рішення. Можливість отримання згоди після впевнення всіх представників підприємства	Виставки, семінари, показові презентації	Дистанційність, надійність, простота	Показати чому реалізація даної системи необхідна для підприємства	Презентація з описом вже існуючого рішення по автоматизації

В результаті аналізу проведеному в пункті 4, було проаналізовано ринок, конкурентів, та схожі представлені товари. Виявлено сильні та слабкі сторони в порівнянні з альтернативними рішеннями, виокремлено можливості для подальшого розвитку проекту та ризики, які потрібно враховувати перед виходом проекту на ринок збуту. Зроблено висновки щодо конкурентоздатності проекту, стратегії його просування на ринку, плани пошуку потенційних замовників і план продажу проекту.

ВИСНОВОК

В результаті вирішення проблеми створення розподіленої системи автоматизації міської системи водопостачання і водовідведення було знайдено рішення, що задовольняє поставленим вимогам до кінцевого продукту, що може бути реалізовано на будь-якому підприємстві, без необхідності заміни вже існуючого на даний момент обладнання. В ході аналізу існуючих приладів і технологій було підібрано найбільш оптимальну структуру програмно-технічного комплексу, що здатен задовільнити вимоги як до зручності системи, так і до її функціоналу і надійності. В результаті практичної роботи було запрограмовано алгоритми обміну інформації між всіма рівнями і підрівнями системи, розроблено SCADA систему та графічний інтерфейс користувача.

Дана структура проекту є універсальною і може бути використана для роботи з будь-яким обладнанням і їх комбінаціями, що є досить перспективним для виходу даного проекту на ринок, де він може зацікавити велику кількість замовників. Аналіз ринку, конкуренції і цільової аудиторії показав, що даний проект має високу конкурентоспроможність і може існувати на ринку створення розподілених систем автоматизації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Баган Т.Г., Батюк С.Г., Бунь В.П., Ізгорев М.Ю., Олійник С.Ю. Методичні рекомендації щодо оформлення курсових та дипломних проектів.— Київ: Політехніка, 2002.- 40с.
- 2 Баган Т.Г., Кисельов Ю.Э., Бунь В.П. Методичні вказівки до вивчення дисципліни „Проектування систем автоматизації” для студентів кафедри АТЕП, Київ НТУУ „КПІ” – 2002 рік.
- 3 Батюк С.Г. Довідковий посібник з комплексного інженерного розрахунку промислових САР в курсовому і дипломному проектуванні.
- 4 Чуйко Ю.М. Конспект лекцій з основ автоматизації технологічних процесів, 2008 р.
- 5 Положення про організацію дипломного проектування та державної атестації студентів НТУУ "КПІ" / Уклад. В.Ю.Угольніков. За заг.ред. Ю.І.Якименка – К.:ВПК "Політехніка", 2006. – 84 с.
- 6 Капля Е.В., Кузеванов В.С., Шевчук В.П. Моделирование процессов управления в интеллектуальных измерительных системах. – М.: ФИЗМАТЛИТ, 2009. – 512с.
- 7 Шевельов В.Е., Скловська Є.Г., Шевченко Т.Є. Методичні вказівки. Сітьове планування та управління виробництвом. – К.: ВПК “Політехніка”, 1993.
- 8 Deshendran Moodley Ontology driven multi-agent system: an architecture for sensor web applications, University of KwaZulu-Natal, Durban, South Africa, December 2009
- 9 Abecker, A. Bernardi, K. Hinkelmann, O. Kuhn, M. Sintek, and K. DFKI. Toward a technology for organizational memories. Intelligent Systems and Their Applications, IEEE 13(3):40–48, 1998.

- 10 F. Bellifemine, G. Caire, T. Trucco, and G. Rimassa. JADE Programmer's Guide. Italy: CSELT SpA, 2:120–122, 2000.
- 11 Fensel, D. Ontologies and Electronic Commerce. IEEE Intelligent Systems, January/February, 8, 2001.
- 12 Hähnle, R. Tableaux and Related Methods. Handbook of Automated Reasoning / Hähnle, R. – Volume I. Elsevier science, 2001. – 277 с.
- 13 Иващенко А. В. Мультиагентные системы для управления производством в реальном времени, Научно-производственная компания «Разумные решения», 2011
- 14 Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений. 2009. № 2. С. 45-57.
- 15 Лобачев П.В. Современные средства измерения расхода жидкости // Интенсификация действующих систем водоснабжения на основе внедрения новой техники и технологии. М.: МДНТП, 1986 г.
- 16 Бочаров В. А., Маркин В. И. Основы логики: Учебник. — М.: ИНФРА-М, 2001. — 296 с
- 17 Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. — Х.: ХНАМГ, 2010. — 222 с
- 18 Гайдамакин Н. А. Автоматизированные системы, базы и банки данных. Вводный курс : Учебное пособие. — М.: Гелиос АРВ, 2002. — 368 с.
- 19 Безгубова Ю.О. Мультиагентное управление распределенными информационными потоками // Образовательные ресурсы и технологии. 2015. № 1(9). С. 113-119.
- 20 Амелин К.С., Баклановский М.В., Граничин О.Н. и др. Адаптивная мультиагентная операционная система реального времени // Стохастическая оптимизация в информатике. 2013. Т. 9. Вып. 1. С. 3-16.

- 21 Брускин Д.Э., Зохорович А.Е., Хвостов В.С. Электрические машины. Часть 1, 2. М.: Высшая школа, 1987 г.
- 22 Сиволов Г.Е., Кармалов А.И., Ивансон П.Б., Исхаков Ю.Б. Многоуровневая автоматизированная система управления технологическими процессами водоснабжения и водоотведения // Водоснабжение и санитарная техника. 2011 г. № 9—1.
- 23 Чебанов В.Б. Стабилизация уровня в приемных резервуарах канализационных насосных станций // Автоматизация и управление систем водоснабжения и водоотведения: Сб. научн. трудов. М.: ВНИИ ВОДГЕО, 1986 г.
- 24 Журба М.Г., Соколов Л.И., Говорова Ж.М. Водоснабжение. Проектирование систем и сооружений. Том 1. М.: Издательство АСВ, 2003
- 25 Порядин А.Ф. Устройство и эксплуатация водозаборов. М.: Стройиздат., 1984.
- 26 Плотников Н.А., Алексеев В.С. Проектирование и эксплуатация водозаборов подземных вод. М.: Стройиздат, 1990 г.
- 27 Гидрология и гидротехнические сооружения: Учеб. для вузов по спец. "Водоснабжение и канализация" / Под ред. Г.Н. Смирнова. - М.: Высш. шк., 1988 г.
- 28 Водозаборно - очистные сооружения и устройства: Учеб. пособие для студентов вузов / Под ред. М.Г. Журбы. - М.: ООО "Издательство Астрель": ООО "Издательство АСТ", 2003 г.
- 29 СНиП 2.01.53-84 "Световая маскировка населенных пунктов и объектов народного хозяйства"
- 30 "Инструкция по категорированию объектов народного хозяйства".
- 31 Оборудование водопроводноканализационных сооружений: Справочник монтажника / Под ред. А.С. Москвитина. М.: Стройиздат, 1980 г.

- 32 Пособие по водоснабжению и канализации городских и сельских поселений (к СНИП 2.07.01-89) // ЦНИИЭП инженерного оборудования, ЦИТП, 1992 г
- 33 Маслов В.П. Інформаційні системи і технології в економіці : Посібник для студ. вузів/ В.П. Маслов; М-во освіти і науки України. -К.: Слово, 2005. -263 с.
- 34 Методи та засоби мультимедійних інформаційних систем : навч. посіб. / Т. М. Басюк, П. І. Жежнич; Нац. ун-т "Львів. політехніка". - Львів : Вид-во Львів. політехніки, 2015. - 426 с. - Бібліогр.: с. 413-416.
- 35 Основи інформаційних систем : Навч. посіб./ Віктор Ситник, Тамара Писаревсь-ка, Ніна Єрьоміна та ін.; За ред. В.Ф.Ситника; М-во освіти України. Київський нац. еко-ном. ун-т. -2-е вид., перероб. і доп.. -К.: КНЕУ, 2001. -420 с.
- 36 Проектування інформаційних систем : Навч. посібник/ Ред. Володимир Пономаренко,. -К.: Академія, 2002. -486 с
- 37 Тиль, П. От нуля к единице : как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с.
- 38 Экланд С. Ангелы, драконы и стервятники : как привлечь правильных инвесторов в свой стартап и сохранить бизнес / С. Экланд ; пер. с англ. О. Терентьевой. – Москва : Манн, Иванов и Фербер, 2011. – 275 с.

Додаток 1

Приклад реалізації алгоритму керування підвищувальною насосною станцією в програмному забезпеченні CoDeSys для ОВЕН ПЛК73, що працює по алгоритму пікового резервування з використанням частотного перетворювача Frecon FR500A та трьох насосів.

Основний алгоритм роботи програми:

```
(*#####действия при запуске*)
```

```
IF bStarting THEN
```

```
(*-----
```

```
----- экран*)
```

```
    ShowString(0,0,0,'ЭКРАНЫ    1из3');
```

```
    ShowString(0,0,1,'основной экран');
```

```
    ShowString(0,0,3,'    войти');
```

```
    ShowString(1,0,0,'ЭКРАНЫ    2из3');
```

```
    ShowString(1,0,1,'журнал аварий');
```

```
    ShowString(1,0,3,'    войти');
```

```
    ShowString(2,0,0,'ЭКРАНЫ    3из3');
```

```
    ShowString(2,0,1,'журнал событий');
```

```
    ShowString(2,0,3,'    войти');
```

```
    Switch_To_Single_Mode();
```

```
(*-----
```

```
----- первичная инициация переменных*)
```

```
    prgAlarmTextFill();
```

```
    prgEventTextFill();
```



```

    IF bSystemRun AND bRestartMode_mO AND bRunModeAuto THEN
bRestartMode:=TRUE; END_IF

```

```

(*-----

```

```

----- считывание данных энергонезависимой памяти*)

```

```

    rlN1WorkTime_mI:=rlN1WorkTime;

```

```

    rlN2WorkTime_mI:=rlN2WorkTime;

```

```

    rlN3WorkTime_mI:=rlN3WorkTime;

```

```

(*-----

```

```

----- выход*)

```

```

    bStarting:=NOT bStarting;

```

```

END_IF

```

```

IF dwSysClock<>0 THEN(*задержка на обновление переменной времени при
включении контроллера*)

```

```

(*#####

```

```

##### Modbus,

```

```

обработка переменных*)

```

```

    prgModbus();

```

```

    bN1chONok:=DI1;

```

```

    bN1ppONok:=DI2;

```

```

    bN2chONok:=DI3;

```

```

    bN2ppONok:=DI4;

```

```

    bN3chONok:=DI5;

```

```

    bN3ppONok:=DI6;

```

```

    rlPressureValue:=funcGetUniSensorValue(rlPressureSensor,rlSensorScaleMin_
mI,rlSensorScaleMax_mI,0.1);

```

```

IF rlN1WorkTime<>rlN1WorkTime_mI THEN
rlN1WorkTime:=rlN1WorkTime_mI; END_IF
IF rlN2WorkTime<>rlN2WorkTime_mI THEN
rlN2WorkTime:=rlN2WorkTime_mI; END_IF
IF rlN3WorkTime<>rlN3WorkTime_mI THEN
rlN3WorkTime:=rlN3WorkTime_mI; END_IF

```

```

(*-----
----- сброс истории аварий*)

```

```

IF bAlarmsHistoryReset_mI AND NOT bAlarmCaution THEN
FOR byReset:=0 TO 31 DO
byAlarmNumber[byReset]:=0;
dwAlarmTime[byReset]:=0;
END_FOR
wAlarmNew:=0;
byAlarmsQty:=0;
bAlarmsHistoryReset_mI:=FALSE;
ELSIF bAlarmsHistoryReset_mI THEN
bAlarmsHistoryReset_mI:=FALSE;
END_IF

```

```

(*-----
----- сброс истории событий*)

```

```

IF bEventsHistoryReset_mI THEN
FOR byReset:=0 TO 31 DO
byEventNumber[byReset]:=0;
dwEventTime[byReset]:=0;
END_FOR
byEventsQty:=0;

```

```

        bEventsHistoryReset_mI:=FALSE;

    END_IF

    (*#####
    #####
    работа системы*)

    trigResetKeyPressed(clk:=wKeyPressed=192);

    IF    (trigResetKeyPressed.q    OR    trigRestartAlarmCheck.q)    AND
bAlarmCaution (*AND NOT bSystemRun*) THEN (*сброс аварий, Alt+Выход*)

        bAlarmCaution:=FALSE;
        bNotAllowAddPump:=FALSE;
        bAlarmStopMode:=FALSE;
        bAlarmStopAll:=FALSE;
        bRunModeAuto:=TRUE;
        prgAlarmsReset();
        bChEmStop:=FALSE;
        bStopModeFRS:=FALSE;
        byChCommandSet:=7;
        wAlarmNew:=0;

    ELSIF    trigResetKeyPressed.q    AND    bAlarmSensorOut    AND    NOT
bAlarmStopMode THEN

        bAlarmCaution:=FALSE;
        bAlarmSensorOut:=FALSE;

    END_IF

    (*-----
    -----*)

    prgUI(); (*интерфейс пользователя*)

```

```

IF byNa<>0 OR byNb<>0 OR byNc<>0 THEN
    bSystemRun:=TRUE;
ELSE
    bSystemRun:=FALSE;
    strRunMessage:='STOP';
    byWorkMode:=0;
END_IF

IF wKeyPressed=32 OR wKPressed=1 THEN
    bSystemStarting:=1;
    wKPressed:=0;
ELSE
    bSystemStarting:=0;
END_IF

trigStartKeyPressed(clk:=bSystemStarting=1);

IF trigStartKeyPressed.Q AND (NOT bAlarmCaution OR
                                (bAlarmCaution      AND
                                bAlarmSensorOut AND NOT bAlarmStopMode)
                                ) THEN
    byWorkMode:=SEL(byWorkMode=0,255,1);
    bRestartMode:=FALSE;
END_IF

trigSystemRun(clk:=bSystemRun);

IF      (trigSystemRun.q      AND      bRunModeAuto)      THEN
prgShiftEventHistory();byEventNumber[0]:=8; END_IF

trigSystemStop(clk:=bSystemRun);

```

```

    IF      (trigSystemRun.q      AND      NOT      bRunModeAuto)      THEN
prgShiftEventHistory();byEventNumber[0]:=0; END_IF
    trigSystemStop(clk:=bSystemRun);
    IF  trigSystemStop.q  THEN  prgShiftEventHistory();byEventNumber[0]:=1;
END_IF
(*-----
----- рестарт системы*)
    IF bRestartMode THEN strRunMessage:='перезапуск'; END_IF
    tonRestartIdle(in:=bRestartMode,pt:=t#6s);
    trigRestartAlarmCheck(clk:=tonRestartIdle.q);
    tonRestartAlarmCheck(in:=tonRestartIdle.q      AND      NOT
trigRestartAlarmCheck.q,pt:=t#3s);
    IF tonRestartAlarmCheck.q AND rlChOut=0 THEN
        bRestartMode:=FALSE;
        IF NOT bAlarmCaution THEN
            prgShiftEventHistory();byEventNumber[0]:=4;
            byWorkMode:=1;
        END_IF
    END_IF
(*=====
=====
===== alarms, events, обработка*)
    IF NOT bAlarmCheckDelay THEN prgAlarms(); END_IF
    prgEvents();
(*=====
=====
===== основной цикл*)
    trigWorkModeIn3(clk:=byWorkMode=3);

```

```

    trigWorkModeIn4(clk:=byWorkMode=4);
    trigWorkModeIn5(clk:=byWorkMode=5);

CASE byWorkMode OF
(*-----
----- старт системы*)

    1:
IF NOT bNotAllowSystemRun THEN
    IF byNa=0 THEN
        prgNx();
        IF byNx<>0 THEN
            byNa:=byNx;
            rlNAchON:=rlNXchON;
            rlNAppON:=rlNXppON;
            bNAchONok:=bNXchONok;
            bNAppONok:=bNXppONok;
        END_IF
    ELSE
        rlNAchON^:=1;
        IF bNAchONok^ THEN
            byChCommandSet:=1;
            rlChSP:=rlChOutMin_mI;
            IF rlChOut>=rlChOutMin_mI-0.1 THEN (*с учетом
гистерезиса*)
                bPIDreset:=TRUE;
                byWorkMode:=2;
            END_IF
        END_IF
    END_IF
END_IF

```

```

END_IF
END_IF
(*-----
----- работа регулятора*)

2:

trigRunModeManual(clk:=NOT bRunModeAuto);

IF bRunModeAuto THEN
    strRunMessage:='автопилот';
    pidRegulator(
        ACTUAL:=rlPressureValue,
        SET_POINT:=rlPressureSP,
        KP:=rlKp_mI,
        TN:=rlTi_mI,
        TV:=0,
        Y_MANUAL:=0,
        Y_OFFSET:=0,
        Y_MIN:=0,
        Y_MAX:=0,
        MANUAL:=FALSE,
        RESET:=bPIDreset);

    IF bPIDreset=TRUE THEN
        bPIDreset:=FALSE;
        rlChSPm:=rlChSP;
        rlPIDyM:=0;
    END_IF

```

```

            IF      ABS(rlPressureValue-rlPressureSP)<0.05      THEN
bPIDreset:=TRUE; END_IF

            tonPIDimpl(in:=NOT      tonPIDimpl.Q      AND      NOT
bPIDreset,PT:=WORD_TO_TIME(wPIDimpl));

            IF tonPIDimpl.q THEN

rlChSP:=rlChSPm+rlPIDyM*(WORD_TO_REAL(wPIDimpl)/1000);

            IF ABS(rlChSP-rlChSPm)<=0.2 THEN
                wPIDimpl:=600;
            ELSE
                wPIDimpl:=200;
            END_IF

            rlPIDyM:=pidRegulator.Y;
            (*IF rlChSP>60 THEN rlChSP:=60;
            ELSIF rlChSP<0 THEN rlChSP:=0;
            ELSE*)
            rlChSPm:=rlChSP;
            (*END_IF*)

            END_IF

            IF  rlChSP>60  AND  ABS(rlPressureValue-rlPressureSP)<0.05
THEN rlChSP:=50; END_IF

            IF  rlChSP>=50+rlChOver_mI  AND  NOT  (rlChOut<50)  AND
NOT bNotAllowedAddPump THEN

                byWorkMode:=3;

```



```

        ELSIF rlChSP<=rlChOutMin_mI-rlChOver_mI AND NOT
(rlChOut>rlChOutMin_mI) THEN
            byWorkMode:=4;
        END_IF

```

```

ELSE

```

```

    IF trigRunModeManual.q THEN
        rlChSP:=rlChOut;
        IF rlChSP<rlChOutMin_mI THEN
            rlChSP:=rlChOutMin_mI;
        ELSIF rlChSP>50 THEN
            rlChSP:=50;
        END_IF

```

```

        rlChSP:=WORD_TO_REAL(REAL_TO_WORD(rlChSP*10))/10;
        bPIDreset:=TRUE;
    END_IF

```

```

END_IF

```

```

(*-----
----- добавление насоса*)

```

```

3:

```

```

IF byNx<>0 AND NOT bNotAllowAddPump THEN
    bStopModeFRS:=TRUE;

```

```

strRunMessage:='начос+';
IF bStopModeFRScheck THEN byChCommandSet:=6; END_IF
(*
IF rlChOut=0 THEN rlNAchON^:=0; END_IF*)
IF                                trigWorkModeIn3.q                                THEN
prgShiftEventHistory();byEventNumber[0]:=2; prgNx(); END_IF
trigNAon(clk:=rlChOut=0);

tonPPon(in:=rlChOut=0,pt:=REAL_TO_TIME(rlSwitchTime_mI*1000));(*зад
ержка при переключении с ПЧ на ПП*)
IF rlChOut=0 THEN
    IF trigNAon.q THEN
        IF byNb=0 THEN
            byNb:=byNa;
            rlNBchON:=rlNAchON;
            rlNBppON:=rlNAppON;
            bNBchONok:=bNAchONok;
            bNBppONok:=bNAppONok;

            byNa:=byNx;
            rlNAchON:=rlNXchON;
            rlNAppON:=rlNXppON;
            bNAchONok:=bNXchONok;
            bNAppONok:=bNXppONok;

            ELSIF byNc=0 THEN
                byNc:=byNb;
                rlNCchON:=rlNBchON;
                rlNCppON:=rlNBppON;

```

```

bNCchONok:=bNBchONok;
bNCppONok:=bNBppONok;

byNb:=byNa;
rlNBchON:=rlNAchON;
rlNBppON:=rlNAppON;
bNBchONok:=bNAchONok;
bNBppONok:=bNAppONok;

byNa:=byNx;
rlNAchON:=rlNXchON;
rlNAppON:=rlNXppON;
bNAchONok:=bNXchONok;
bNAppONok:=bNXppONok;
END_IF
END_IF

tonChOnDelay(in:=tonPPon.q,PT:=t#1500ms);
IF tonPPon.q THEN
    rlNBchON^:=0;
    IF NOT bNBchONok^ THEN rlNBppON^:=1;
END_IF

    bStopModeFRS:=FALSE;
    IF tonChOnDelay.Q THEN byWorkMode:=1;
END_IF

END_IF
END_IF
ELSE

```

```

        rlChSP:=50;
        bPIDreset:=TRUE;
        byWorkMode:=2;
    END_IF
IF byNx<>0 AND bNotAllowAddPump THEN
    rlChSP:=50;
    bPIDreset:=TRUE;
    byWorkMode:=2;
END_IF

(*-----
----- ВЫКЛЮЧЕНИЕ насоса*)

4:

    IF trigWorkModeIn4.q THEN
        prgShiftEventHistory();byEventNumber[0]:=3;
    END_IF
    strRunMessage:='насос-';
    IF byNc<>0 OR byNb<>0 AND NOT bAlarmN1ContPPoff AND NOT
bAlarmN2ContPPoff THEN
        rlChSP:=50;
        IF rlChOut>40 THEN
            IF byNc<>0 THEN
                rlNCppON^:=0;
                prgResetPointerC();
            ELSIF byNb<>0 THEN
                rlNBppON^:=0;
                prgResetPointerB();

```

```

        END_IF
        bPIDreset:=TRUE;
        byWorkMode:=2;
    END_IF
ELSE
    IF bSleepMode_mO THEN
        byChCommandSet:=5;
        IF rlChOut=0 THEN byWorkMode:=5; END_IF
    ELSE
        rlChSP:=rlChOutMin_mI;
        bPIDreset:=TRUE;
        byWorkMode:=2;
    END_IF
    (*IF rlChSP<24.5 THEN
        byChCommandSet:=6;
        IF rlChOut=0 THEN byWorkMode:=5; END_IF
    ELSE
        rlChSP:=rlChOutMin_mI;
        bPIDreset:=TRUE;
        byWorkMode:=2;
    END_IF*)
END_IF
(*-----
----- coh*)

5:

IF trigWorkModeIn5.q THEN
    prgShiftEventHistory();byEventNumber[0]:=5;

```

END_IF

strRunMessage:='coH';

IF rlPressureValue<rlPressureSP-rlSleepOffDelta_mI THEN

prgShiftEventHistory();byEventNumber[0]:=5;

byWorkMode:=1;

END_IF

(*-----
----- остановка*)

255:

IF byNc<>0 THEN

rlNCppON^:=0;

tonPPoff(in:=NOT bNCppONok^,pt:=t#3s);

strRunMessage:=CONCAT(funcRealToString((3000-
TIME_TO_WORD(tonPPoff.et))/1000,1),'cek');

IF tonPPoff.Q THEN prgResetPointerC(); END_IF

ELSIF byNb<>0 THEN

rlNBppON^:=0;

tonPPoff(in:=NOT bNBppONok^,pt:=t#3s);

strRunMessage:=CONCAT(funcRealToString((3000-
TIME_TO_WORD(tonPPoff.et))/1000,1),'cek');

IF tonPPoff.Q THEN prgResetPointerB(); END_IF

ELSIF byNa<>0 THEN

byChCommandSet:=5;

IF rlChOut=0 THEN

rlNAchON^:=0;

prgResetPointerA();

```

END_IF
END_IF
END_CASE
(*#####
#####
##### мотоцисы*)
    tonN1Timer(in:=NOT    tonN1Timer.q    AND    (bN1chONok    OR
bN1ppONok),PT:=t#180s);
    IF                                tonN1Timer.Q                                THEN
rlN1WorkTime_mI:=DWORD_TO_REAL(REAL_TO_DWORD(rlN1WorkTime
_mI*100)+5)/100; END_IF

    tonN2Timer(in:=NOT    tonN2Timer.q    AND    (bN2chONok    OR
bN2ppONok),PT:=t#180s);
    IF                                tonN2Timer.Q                                THEN
rlN2WorkTime_mI:=DWORD_TO_REAL(REAL_TO_DWORD(rlN2WorkTime
_mI*100)+5)/100; END_IF

    tonN3Timer(in:=NOT    tonN3Timer.q    AND    (bN3chONok    OR
bN3ppONok),PT:=t#180s);
    IF                                tonN3Timer.Q                                THEN
rlN3WorkTime_mI:=DWORD_TO_REAL(REAL_TO_DWORD(rlN3WorkTime
_mI*100)+5)/100; END_IF

(*#####Екстренное переключение насосов*)

tofGetAnotherPump1(in:=bN1auto, PT:=t#15s);

```

```

IF bN1chONok=1 AND bN1auto = 0 THEN
byChCommandSet:=5;
IF rlChOut=0 AND bN2auto=0 THEN
prgResetPointerA();
prgResetPointerB();
rlN1ppON:=0;
rlN1chON:=0;
rlN2ppON:=0;
rlN2chON:=0;
END_IF
IF rlChOut=0 AND bN2auto=1 THEN
rlN1chON:=0;
prgResetPointerA();
prgResetPointerB();
rlN2ppON:=0;
prgResetPointerA();
prgResetPointerB();
byWorkMode:=0;
END_IF
END_IF
IF byWorkMode = 0 THEN
bDelay1:= FALSE;
ELSE
bDelay1:=TRUE;
END_IF
tofGetAnotherPump1(in:=bDelay1, PT:=t#2s);
IF tofGetAnotherPump1.Q = 0 AND bN2auto=1 THEN
wKPressed:=1;

```


END_IF

IF bN2chONok=1 AND bN2auto = 0 THEN

byChCommandSet:=5;

IF rlChOut=0 AND bN1auto=0 THEN

prgResetPointerA();

prgResetPointerB();

rlN1ppON:=0;

rlN1chON:=0;

rlN2ppON:=0;

rlN2chON:=0;

END_IF

IF rlChOut=0 AND bN1auto=1 THEN

rlN2chON:=0;

prgResetPointerA();

prgResetPointerB();

rlN1ppON:=0;

prgResetPointerA();

prgResetPointerB();

byWorkMode:=0;

END_IF

END_IF

IF byWorkMode = 0 THEN

bDelay2:= FALSE;

ELSE

bDelay2:=TRUE;

END_IF

```
tofGetAnotherPump2(in:=bDelay2, PT:=t#2s);  
IF tofGetAnotherPump2.Q = 0 AND bN1auto=1 THEN  
wKPressed:=1;  
END_IF
```

```
IF bN1ppONok=1 AND bN1auto = 0 THEN  
rlN1ppON:=0;  
END_IF  
IF bN2ppONok=1 AND bN2auto = 0 THEN  
rlN2ppON:=0;  
END_IF
```

```
tonCavitStop(in:=bCavitStop, pt:=t#4m);
```

```
rlCurrentOut:=rlCurrentOut/10;  
IF rlCurrentOut-rlCavitCurrent<1 THEN  
bCavitStop:=TRUE;  
ELSE  
bCavitStop:=FALSE;  
END_IF
```

```
IF tonCavitStop.Q THEN  
IF bN1ppONok = 1 THEN rlN1ppON:=0; bCavitStop:=FALSE;  
ELSIF bN2ppONok = 1 THEN rlN2ppON:=0; bCavitStop:=FALSE;  
END_IF  
END_IF
```

IF bN1auto AND NOT bHA1State THEN

 bHA1State:=TRUE;

 prgShiftEventHistory();byEventNumber[0]:=9;

END_IF

IF NOT bN1auto AND bHA1State THEN

 bHA1State:=FALSE;

 prgShiftEventHistory();byEventNumber[0]:=10;

END_IF

IF bN2auto AND NOT bHA2State THEN

 bHA2State:=TRUE;

 prgShiftEventHistory();byEventNumber[0]:=11;

END_IF

IF NOT bN2auto AND bHA2State THEN

 bHA2State:=FALSE;

 prgShiftEventHistory();byEventNumber[0]:=12;

END_IF

IF NOT bN1auto AND bN1ppONok AND NOT bH1ppState THEN

 bH1ppState:=TRUE;

 prgShiftEventHistory();byEventNumber[0]:=13;

END_IF

IF NOT bN2auto AND bN2ppONok AND NOT bH2ppState THEN

 bH2ppState:=TRUE;

 prgShiftEventHistory();byEventNumber[0]:=14;

END_IF

IF NOT bN1auto AND NOT bN1ppONok AND bH1ppState THEN

 bH1ppState:=FALSE;

 prgShiftEventHistory();byEventNumber[0]:=15;

END_IF

IF NOT bN2auto AND NOT bN2ppONok AND bH2ppState THEN

 bH2ppState:=FALSE;

 prgShiftEventHistory();byEventNumber[0]:=16;

END_IF

END_IF

Алгоритм ініціалізації і реагування на аварійні ситуації:

(*#####

анализ аварий*)

(*-----
----- (!) аварийный останов оператором*)

IF NOT bEmStop AND NOT bAlarmOperatorEmStop THEN

 bAlarmOperatorEmStop:=TRUE; bAlarmStopMode:=TRUE;

 prgShiftAlarmHistory(); byAlarmNumber[0]:=0;

END_IF

(*-----
----- (!) авария питания на входе*)

tonPowerOut(in:=NOT bPowerOK, pt:=t#3s);

IF tonPowerOut.q THEN

 bAlarmPowerOut:=TRUE; bAlarmStopMode:=TRUE;

 prgShiftAlarmHistory(); byAlarmNumber[0]:=1;

END_IF

tonPowerIn(in:=bPowerOk, pt:=t#3s);

IF tonPowerIn.Q AND bAlarmPowerOut AND bAlarmStopMode AND NOT
bAlarmOperatorEmStop AND NOT bSuhoyHod AND bEmStop AND NOT
bAlarmSensorOut AND NOT bAlarmMK110 AND NOT bAlarmChOwn AND
NOT bAlarmChLink AND NOT bAlarmChContactors THEN

bAlarmStopMode:=FALSE;

bRunModeAuto:=TRUE;

byWorkMode:=1;

END_IF

(*-----
----- (!) обрыв связи с МК110*)

tonLinkErrorDev16(in:=intMbErrorDev16<>0, pt:=tLinkCheckIdle);

IF tonLinkErrorDev16.Q AND NOT bAlarmMK110 THEN

 bAlarmMK110:=TRUE; bAlarmStopMode:=TRUE;

 prgShiftAlarmHistory(); byAlarmNumber[0]:=2;

END_IF

(*=====

=====

=====*)

(*-----
----- авария ПЧ*)

tonChReset(in:=NOT bAlarmChOwn, pt:=t#1000ms); (*задержка, чтоб reset
подействовало*)

(*tonGetAlarmCause(in:=wChOutputs.0, pt:=t#200ms); (*задержка, чтоб считать
причину аварии*)*)

IF WORD_TO_BYTE(wChTripCouse)<>0 AND NOT tonLinkErrorDev1.Q AND
tonChReset.q (*AND tonGetAlarmCause.q*) AND NOT bAlarmChOwn THEN

 bAlarmChOwn:=TRUE;

 bAlarmStopMode:=TRUE;

 prgShiftAlarmHistory();

byAlarmNumber[0]:=100+WORD_TO_BYTE(wChTripCouse);

END_IF

(*-----
----- обрыв связи с ПЧ*)

tonLinkErrorDev1(in:=intMbErrorDev1<>0, pt:=tLinkCheckIdle);

IF tonLinkErrorDev1.Q AND NOT bAlarmChLink THEN

 bAlarmChLink:=TRUE; bAlarmStopMode:=TRUE;

 prgShiftAlarmHistory(); byAlarmNumber[0]:=3;

END_IF

IF NOT tonLinkErrorDev1.Q AND bAlarmChLink THEN

 bAlarmChLink:=FALSE; bAlarmStopMode:=FALSE;

prgAlarmsReset();

byWorkMode:=1;

END_IF

(*=====

=====

=====*)

(*-----
----- не сработал контактор ПП*)

tonContactorN1ppOff (in:=rlN1ppON=1, pt:=tContactorCheckIdle);

IF tonContactorN1ppOff.q AND NOT bN1ppONok AND NOT
bAlarmN1ContPPoff THEN

 bAlarmN1ContPPoff:=TRUE;

 (*bAlarmStopMode:=TRUE;*)

 prgShiftAlarmHistory(); byAlarmNumber[0]:=5;

 (*IF NOT bNAchONok^ AND byNc=0 THEN bAlarmKeepMode:=TRUE;

END_IF*)

END_IF

```

tonContactorNotAllow1 (in:=bAlarmN1ContPPoff, pt:=t#10s);
    IF tonContactorNotAllow1.q THEN
        bNotAllowAddPump:=TRUE;
        rlN1ppON:=0;
    END_IF
tonContactorN2ppOff (in:=rlN2ppON=1, pt:=tContactorCheckIdle);
IF  tonContactorN2ppOff.q  AND  NOT  bN2ppONok  AND  NOT
bAlarmN2ContPPoff THEN
    bAlarmN2ContPPoff:=TRUE;
    (*bAlarmStopMode:=TRUE;*)
    prgShiftAlarmHistory(); byAlarmNumber[0]:=6;

    (*IF NOT bNAchONok^ AND byNc=0 THEN bAlarmKeepMode:=TRUE;
END_IF*)
END_IF
    tonContactorNotAllow2 (in:=bAlarmN2ContPPoff, pt:=t#10s);
    IF tonContactorNotAllow2.q THEN
        bNotAllowAddPump:=TRUE;
        rlN2ppON:=0;
    END_IF
tonContactorN3ppOff (in:=rlN3ppON=1, pt:=tContactorCheckIdle);
IF  tonContactorN3ppOff.q  AND  NOT  bN3ppONok  AND  NOT
bAlarmN3ContPPoff THEN
    bAlarmN3ContPPoff:=TRUE;
    (*bAlarmStopMode:=TRUE;*)
    prgShiftAlarmHistory(); byAlarmNumber[0]:=7;
    (*IF NOT bNAchONok^ AND byNc=0 THEN bAlarmKeepMode:=TRUE;
END_IF*)

```

END_IF

tonContactorNotAllow3 (in:=bAlarmN3ContPPoff, pt:=t#10s);

IF tonContactorNotAllow3.q THEN

bNotAllowAddPump:=TRUE;

rlN3ppON:=0;

END_IF

(*-----
----- не сработал контактор ПЧ*)

tonContactorN1chOff (in:=rlN1chON=1, pt:=tContactorCheckIdle);

IF tonContactorN1chOff.q AND NOT bN1chONok AND NOT
bAlarmN1ContChoff THEN

bAlarmN1ContChoff:=TRUE; bAlarmStopMode:=TRUE;

prgShiftAlarmHistory(); byAlarmNumber[0]:=8;

END_IF

tonContactorN2chOff (in:=rlN2chON=1, pt:=tContactorCheckIdle);

IF tonContactorN2chOff.q AND NOT bN2chONok AND NOT
bAlarmN2ContChoff THEN

bAlarmN2ContChoff:=TRUE; bAlarmStopMode:=TRUE;

prgShiftAlarmHistory(); byAlarmNumber[0]:=9;

END_IF

tonContactorN3chOff (in:=rlN3chON=1, pt:=tContactorCheckIdle);

IF tonContactorN3chOff.q AND NOT bN3chONok AND NOT
bAlarmN3ContChoff THEN

bAlarmN3ContChoff:=TRUE; bAlarmStopMode:=TRUE;

prgShiftAlarmHistory(); byAlarmNumber[0]:=10;

END_IF

(*-----
----- залип контактор ПП*)


```

tonContactorN1ppOn (in:=r1N1ppON=0, pt:=tContactorCheckIdle);
IF tonContactorN1ppOn.q AND bN1ppONok AND bSystemRun AND NOT
bAlarmN1ContPPon AND bN1auto THEN (*bN1auto чтоб не срабатывало в
авар. ПП*)
    bAlarmN1ContPPon:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=11;
ELSIF tonContactorN1ppOn.q AND bN1ppONok AND bSystemRun AND NOT
bAlarmN1ContPPon AND NOT bN1auto AND NOT bAlarmStopAll THEN
    bAlarmStopAll:=TRUE; bAlarmStopMode:=TRUE; bSystemRun:=FALSE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=20; bDiffMode1:=TRUE;
    byWorkMode:=255;
END_IF
tonDiffMode1(in:=bDiffMode1, pt:=t#5s);
IF tonDiffMode1.Q THEN
    r1N1ppON:=0;
    r1N1chON:=0;
    r1N2ppON:=0;
    r1N2chON:=0;
END_IF
tonContactorN2ppOn (in:=r1N2ppON=0, pt:=tContactorCheckIdle);
IF tonContactorN2ppOn.q AND bN2ppONok AND bSystemRun AND NOT
bAlarmN2ContPPon AND bN2auto THEN
    bAlarmN2ContPPon:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=12;
ELSIF tonContactorN2ppOn.q AND bN2ppONok AND bSystemRun AND NOT
bAlarmN2ContPPon AND NOT bN2auto AND NOT bAlarmStopAll THEN
    bAlarmStopAll:=TRUE; bAlarmStopMode:=TRUE; bSystemRun:=FALSE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=20; bDiffMode2:=TRUE;

```

```

    byWorkMode:=255;
END_IF
tonDiffMode2(in:=bDiffMode2, pt:=t#5s);
IF tonDiffMode2.Q THEN
    rlN1ppON:=0;
    rlN1chON:=0;
    rlN2ppON:=0;
    rlN2chON:=0;
END_IF
tonContactorN3ppOn (in:=rlN3ppON=0, pt:=tContactorCheckIdle);
IF tonContactorN3ppOn.q AND bN3ppONok AND bSystemRun AND NOT
bAlarmN3ContPPon AND bN3auto THEN
    bAlarmN3ContPPon:=TRUE;                                bAlarmStopMode:=TRUE;
bSystemRun:=FALSE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=13;
ELSIF tonContactorN3ppOn.q AND bN3ppONok AND bSystemRun AND NOT
bAlarmN3ContPPon AND NOT bN3auto AND NOT bAlarmStopAll THEN
    bAlarmStopAll:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=20;
    rlN1ppON:=0;
    rlN1chON:=0;
    rlN2ppON:=0;
    rlN2chON:=0;
    byWorkMode:=255;
END_IF
(*-----
----- залип контактор ПЧ*)
tonContactorN1chOn (in:=rlN1chON=0, pt:=tContactorCheckIdle);

```

```

IF tonContactorN1chOn.q AND bN1chONok AND NOT bAlarmN1ContChon
THEN
    bAlarmN1ContChon:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=14;
END_IF
tonContactorN2chOn (in:=rlN2chON=0, pt:=tContactorCheckIdle);
IF tonContactorN2chOn.q AND bN2chONok AND NOT bAlarmN2ContChon
THEN
    bAlarmN2ContChon:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=15;
END_IF
tonContactorN3chOn (in:=rlN3chON=0, pt:=tContactorCheckIdle);
IF tonContactorN3chOn.q AND bN3chONok AND NOT bAlarmN3ContChon
THEN
    bAlarmN3ContChon:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=16;
END_IF
(*-----
----- совместное включение контакторов ПЧ*)
IF ((bN1chONok AND bN2chONok) OR (bN2chONok AND bN3chONok) OR
(bN1chONok AND bN3chONok)) AND NOT bAlarmChContactors THEN
    bAlarmChContactors:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=17;
END_IF
(*-----
----- встречное включение ПП и ПЧ*)
IF bN1chONok AND bN1ppONok AND NOT bAlarmN1Contactors THEN
    bAlarmN1Contactors:=TRUE; bAlarmStopMode:=TRUE;

```

```

    prgShiftAlarmHistory(); byAlarmNumber[0]:=18;
END_IF
IF bN2chONok AND bN2ppONok AND NOT bAlarmN2Contactors THEN
    bAlarmN2Contactors:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=19;
END_IF
IF bN3chONok AND bN3ppONok AND NOT bAlarmN3Contactors THEN
    bAlarmN3Contactors:=TRUE; bAlarmStopMode:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=20;
END_IF
(*=====
=====
=====*)
(*-----
----- сух. ход*)
tonSuhoyHod(in:= bSuhoyHod, pt:=t#1s);
IF tonSuhoyHod.Q AND NOT bAlarmSuhoyHod THEN
    bAlarmSuhoyHod:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=4;
END_IF
(*-----
----- обрыв датчика*)
IF (rlPressureSensor<-1 OR rlPressureSensor>100) AND NOT bAlarmSensorOut
THEN (*3% запаса для избежания срабатывания аварии на границах*)
    bAlarmSensorOut:=TRUE; bAlarmStopAll:=TRUE; byWorkMode:=255;
    bNotAllowSystemRun:=TRUE;
    prgShiftAlarmHistory(); byAlarmNumber[0]:=21;
END_IF

```

```
IF rIPressureSensor>-1 AND rIPressureSensor<100 AND bAlarmSensorOut THEN
(*3% запаса для избежания срабатывания аварии на границах*)
```

```
    bAlarmSensorOut:=FALSE; (*bAlarmStopAll:=FALSE; byWorkMode:=1;*)
    bNotAllowSystemRun:=FALSE; byWorkMode:=1;
```

```
END_IF
```

```
(*#####
#####
```

```
реакции на аварии*)
```

```
(*IF    bAlarmCaution    AND    NOT    bAlarmN1ContPPoff    AND    NOT
bAlarmN2ContPPoff    AND    NOT    bAlarmN3ContPPoff    THEN
bRunModeAuto:=FALSE; END_IF*)
```

```
IF bAlarmStopMode THEN
```

```
    byWorkMode:=0;
```

```
    IF NOT bN1chONok AND NOT bN1ppONok THEN
```

```
        IF            byNa=1    AND    NOT    bAlarmKeepMode    THEN
prgResetPointerA();
```

```
        ELSIF        byNb=1 THEN prgResetPointerB();
```

```
        ELSIF        byNc=1 THEN prgResetPointerC();
```

```
        END_IF
```

```
    END_IF
```

```
    IF NOT bN2chONok AND NOT bN2ppONok THEN
```

```
        IF            byNa=2    AND    NOT    bAlarmKeepMode    THEN
prgResetPointerA();
```

```
        ELSIF        byNb=2 THEN prgResetPointerB();
```

```
        ELSIF        byNc=2 THEN prgResetPointerC();
```

```

        END_IF
    END_IF
    IF NOT bN3chONok AND NOT bN3ppONok THEN
        IF      byNa=3    AND    NOT    bAlarmKeepMode    THEN
prgResetPointerA();
        ELSIF    byNb=3 THEN prgResetPointerB();
        ELSIF    byNc=3 THEN prgResetPointerC();
        END_IF
    END_IF

    IF    bSystemRun AND NOT bAlarmStopAll AND
        NOT (bAlarmOperatorEmStop OR bAlarmPowerOut OR bAlarmMK110
OR bAlarmChLink) (*исключение надписи в критических авариях*)
    THEN strRunMessage:='удержание'; END_IF

    IF wKeyPressed=32 THEN bAlarmStopAll:=TRUE; END_IF (*условия
полной остановки*)

    IF bAlarmStopAll THEN

        rlN1ppON:=0;
        rlN2ppON:=0;
        rlN3ppON:=0;

        byChCommandSet:=6;
        IF rlChOut=0 THEN
            rlN1chON:=0;
            rlN2chON:=0;

```

```

        rlN3chON:=0;

    END_IF

END_IF

END_IF

(*----- экстренная остановка
оператором, авария питания на входе, обрывы связи с МК110/ПЧ*)
IF bAlarmOperatorEmStop OR (bAlarmPowerOut AND NOT tonPowerIn.Q) OR
bAlarmMK110 OR bAlarmChLink THEN
    rlN1ppON:=0;
    rlN2ppON:=0;
    rlN3ppON:=0;
    bChEmStop:=1; byChCommandSet:=6;
    IF bAlarmChOwn OR bAlarmMK110 OR bAlarmChLink THEN
        rlN1chON:=0;
        rlN2chON:=0;
        rlN3chON:=0;
    END_IF
END_IF

(*-----
ПЧ в аварии, совместное срабатывание контакторов ПЧ*)
IF bAlarmChOwn OR bAlarmChContactors THEN
    bChEmStop:=1; byChCommandSet:=6;
    rlN1chON:=0;
    rlN2chON:=0;
    rlN3chON:=0;
END_IF

```

```

(*-----
----- не срабатывание контакторов ПП*)
IF bAlarmN1ContPPoff THEN rlN1ppON:=0; END_IF
IF bAlarmN2ContPPoff THEN rlN2ppON:=0; END_IF
IF bAlarmN3ContPPoff THEN rlN3ppON:=0; END_IF

IF bAlarmKeepMode AND (bAlarmN1ContPPoff OR bAlarmN2ContPPoff OR
bAlarmN3ContPPoff) AND (wAlarmNew=1 OR (wAlarmNew=3 AND
bAlarmSensorOut)) THEN
    rlNAchON^:=1;
    IF bNAchONok^ THEN
        byChCommandSet:=1;
        rlChSP:=50;
        bAlarmKeepMode:=FALSE;
    END_IF
END_IF

(*-----
----- не срабатывание контакторов ПЧ*)
IF bAlarmN1ContChoff THEN
    bChEmStop:=1; byChCommandSet:=6;
    rlN1chON:=0;
END_IF
IF bAlarmN2ContChoff THEN
    bChEmStop:=1; byChCommandSet:=6;
    rlN2chON:=0;
END_IF
IF bAlarmN3ContChoff THEN
    bChEmStop:=1; byChCommandSet:=6;

```



```

    rlN3chON:=0;
END_IF
(*-----
----- встречное срабатывание контакторов ПП и ПЧ*)
IF bAlarmN1Contactors THEN
    bChEmStop:=1; byChCommandSet:=6;
    rlN1ppON:=0;
    rlN1chON:=0;
END_IF
IF bAlarmN2Contactors THEN
    bChEmStop:=1; byChCommandSet:=6;
    rlN2ppON:=0;
    rlN2chON:=0;
END_IF
IF bAlarmN3Contactors THEN
    bChEmStop:=1; byChCommandSet:=6;
    rlN3ppON:=0;
    rlN3chON:=0;
END_IF
(*-----
----- сух. ход*)
IF bAlarmSuhoyHod THEN
    (*IF      byWorkMode<>0      THEN*)      byWorkMode:=255;
    (*bAlarmStopAll:=TRUE; END_IF*)
END_IF

tonSuhoyHodReset(in:=NOT bSuhoyHod, pt:=t#10s);

```

```
IF tonSuhoyHodReset.Q AND bAlarmSuhoyHod THEN bAlarmSuhoyHod:=0;
bAlarmStopAll:=FALSE; byWorkMode:=1; END_IF
```

```
IF wChTripCouse = 18 THEN byChCommandSet:=7; END_IF
```

Реалізація графічного інтерфейсу користувача та керування інформаційними
лампами: trigTheScreens(clk:=Get_Display_Mode()<>SINGLE_MODE);

```
trigAScreenIn(clk:=Get_Display_Mode()=SINGLE_MODE);
```

```
(*#####  
#####  
##### ЭКРАНЫ*)
```

```
IF Get_Display_Mode()<>SINGLE_MODE THEN
```

```
(*=====
```

```
=====
```

```
===== таймер возврата*)
```

```
tonScreenBack(in:=NOT tonScreenBack.q AND wKeyPressed=0, pt:=t#10s);
```

```
IF tonScreenBack.q THEN
```

```
SetWorkScreen(0);
```

```
Switch_To_Single_Mode();
```

```
ELSE
```

```
(*=====
```

```
=====
```

```
===== инициализация*)
```

```
IF trigTheScreens.q THEN
```

```
SetWorkScreen(dintCurrWorkScreen);
```

```
END_IF
```

```

(*=====
=====
===== нажатия*)

    trigKeyPressed(clk:=wKeyPressed<>0);
    IF trigKeyPressed.q AND NOT trigTheScreens.q THEN
        IF wKeyPressed=2 THEN
            Switch_To_Single_Mode();
        END_IF
    END_IF
END_IF

ELSE
(*#####
#####
# базовый экран*)
    IF GetWorkScreen()=0 THEN
(*=====
=====
===== инициализация*)

        BlinkTimer(enable:=TRUE, timelow:=T#600ms,timehigh:=T#600ms);
        IF trigAScreenIn.q THEN
            Clear_Display(Single_Mode);
        END_IF

(*=====
=====
===== экран*)

        (* IF trigScreenIn.q OR tonScreenRefresh.q THEN
            ;
        END_IF

```

*)

(*-----

----- 0-я строка*)

IF bN1auto OR (NOT bN1auto AND byNa=1) THEN

IF bN1chONok THEN strTextLine:='=H1=';

ELSIF bN1ppONok THEN strTextLine:='-H1-';

ELSE strTextLine:=' H1 ';

END_IF

ELSE

strTextLine:=' ';

END_IF

IF bN2auto OR (NOT bN2auto AND byNa=2) THEN

IF bN2chONok THEN

strTextLine:=CONCAT(strTextLine,' =H2=');

ELSIF bN2ppONok THEN

strTextLine:=CONCAT(strTextLine,' -H2-');

ELSE

strTextLine:=CONCAT(strTextLine,' H2 ');

END_IF

ELSE

strTextLine:=CONCAT(strTextLine,' ');

END_IF

IF bN3auto OR (NOT bN3auto AND byNa=3) THEN

IF bN3chONok THEN

strTextLine:=CONCAT(strTextLine,' =H3=');

```

                ELSIF                                bN3ppONok                THEN
strTextLine:=CONCAT(strTextLine,' -H3-');
                ELSE
strTextLine:=CONCAT(strTextLine,' H3 ');
                END_IF
ELSE
                strTextLine:=CONCAT(strTextLine,' ');
END_IF

Write_Display(Single_mode,0,0,strTextLine);

(*-----
----- 1-я строка*)

strTextLine:=CONCAT('Д:',funcRealToString(rlPressureValue,0.1));

bPermChSpChange:=FALSE;
IF bRunModeAuto THEN
    strTextLine:=CONCAT(strTextLine, '/');

strTextLine:=CONCAT(strTextLine,funcRealToString(rlPressureSP,0.1));
    strTextLine:=CONCAT(strTextLine, 'bar');
    ELSIF (NOT bRunModeAuto AND byWorkMode=2) OR
(bAlarmStopMode AND byChCommandSet=1) THEN
        bPermChSpChange:=TRUE;
        strTextLine:=CONCAT(strTextLine, 'bar/');

strTextLine:=CONCAT(strTextLine,funcRealToString(rlChSP,0.1));
        strTextLine:=CONCAT(strTextLine, 'ГЦ');
    ELSE

```

```

        strTextLine:=CONCAT(strTextLine,'bar');
    END_IF

    strTextLine:=funcLineEndsFill(strTextLine,'right');
    Write_Display(Single_mode,0,1,strTextLine);

    (*-----
    ----- 2-я строка: строка сообщений*)
        strRunMessage:=funcLineEndsFill(strRunMessage,'right');
        strRunMessage:=REPLACE(strRunMessage,'Гц',1,15);

        strRunMessage:=REPLACE(strRunMessage,funcRealToString(rlChOut,1),LEN(
funcRealToString(rlChOut,1)),15-LEN(funcRealToString(rlChOut,1)));
        strRunMessage:=REPLACE(strRunMessage,'',1,14-
LEN(funcRealToString(rlChOut,1)));
        Write_Display(Single_mode,0,2,strRunMessage);
        strRunMessage:="";

    (*-----
    ----- 3-я строка: строка меню*)
        strTextLine:='';

    IF NOT bAlarmStopMode THEN
        IF bRunModeAuto THEN
            strTextLine:=REPLACE(strTextLine,'авт.р',5,12);
        ELSIF NOT bRunModeAuto THEN
            strTextLine:=REPLACE(strTextLine,'ПУЧ.р',5,12);
        END_IF
    END_IF

```

```

bPermManualPunpUp:=FALSE;
bPermManualPunpDown:=FALSE;
IF byWorkMode=2 THEN
    prgNx();
    IF NOT bRunModeAuto AND rlChOut=50 AND byNx<>0 THEN

        strTextLine:=REPLACE(strTextLine,'H',1,1);

strTextLine:=REPLACE(strTextLine,BYTE_TO_STRING(byNx),1,2);
        strTextLine:=REPLACE(strTextLine,'+',1,3);
        bPermManualPunpUp:=TRUE;

        ELSIF NOT bRunModeAuto AND rlChOut=rlChOutMin_mI
AND (byNb<>0 OR byNc<>0) THEN

            strTextLine:=REPLACE(strTextLine,'H',1,1);
            IF byNc<>0 THEN

strTextLine:=REPLACE(strTextLine,BYTE_TO_STRING(byNc),1,2);
                ELSIF byNb<>0 THEN

strTextLine:=REPLACE(strTextLine,BYTE_TO_STRING(byNb),1,2);
                    END_IF
                    strTextLine:=REPLACE(strTextLine,'-',1,3);
                    bPermManualPunpDown:=TRUE;

                END_IF
            END_IF

```

```

Write_Display(Single_mode,0,3,strTextLine);

(*=====
=====
===== нажатия*)

    trigKeyPressed(clk:=wKeyPressed<>0);
    trigKeysPressed(clk:=wKeyPressed=132 OR wKeyPressed=136);
    tonKeyLongPress(in:=wKeyPressed=8 OR wKeyPressed=4, pt:=t#1s);
    IF    tonKeyLongPress.q    THEN    tonValueBoost    (in:=NOT
tonValueBoost.Q, pt:=t#300ms); END_IF

    IF (trigKeyPressed.q OR trigKeysPressed.q OR tonKeyLongPress.q)
AND NOT trigAScreenIn.q THEN

        IF    (wKeyPressed=8    OR    wKeyPressed=4)    AND
bPermChSpChange THEN

            IF wKeyPressed=8 THEN rlShift:=1;
            ELSIF wKeyPressed=4 THEN rlShift:=-1;
            END_IF

            IF tonKeyLongPress.q THEN
                IF    tonValueBoost.q    THEN
rlChSP:=rlChSP+0.3*rlShift; END_IF
            ELSE
                rlChSP:=rlChSP+0.1*rlShift;;
            END_IF

```



```

IF rlChSP>50 THEN
    rlChSP:=50;
ELSIF rlChSP<rlChOutMin_mI THEN
    rlChSP:=rlChOutMin_mI;
END_IF

rlChSP:=WORD_TO_REAL(REAL_TO_WORD(rlChSP*10))/10;

ELSIF (wKeyPressed=136 OR wKeyPressed=132) AND
bRunModeAuto THEN

    IF wKeyPressed=136 THEN rlShift:=1;
    ELSIF wKeyPressed=132 THEN rlShift:=-1;
    END_IF

    rlPressureSP:=rlPressureSP+0.1*rlShift;

    rlPressureSP:=WORD_TO_REAL(REAL_TO_WORD(rlPressureSP*10))/10;

ELSIF wKeyPressed=256 AND NOT bAlarmStopMode THEN
(*переключение режимов, F3*)
    bRunModeAuto:=NOT bRunModeAuto;

ELSIF wKeyPressed=1 THEN (*переключение насосов,
Alt+F3*)

    IF bPermManualPunpUp THEN byWorkMode:=3;
    ELSIF bPermManualPunpDown THEN byWorkMode:=4;
    END_IF

```

```

        ELSIF wKeyPressed=64 THEN
            dintCurrWorkScreen:=GetWorkScreen();
            Switch_To_PREV_Mode();
        END_IF
    END_IF

END_IF

(*#####
#####
# журнал аварий*)
    IF GetWorkScreen()=1 THEN
        (*=====
=====
===== таймер возврата*)
            tonScreenBack(in:=NOT tonScreenBack.q AND wKeyPressed=0,
pt:=t#30s);

        IF tonScreenBack.q THEN
            dintCurrWorkScreen:=GetWorkScreen();
            Switch_To_PREV_Mode();
        ELSE
            (*=====
=====
===== инициализация*)
                tonScreenRefresh(in:=NOT tonScreenRefresh.q, pt:=t#200ms);

            IF trigAScreenIn.q THEN

```

```

        byCurrListPosition:=0;
        Clear_Display(Single_mode);
        prgRunLineIni();
    END_IF

    (*=====
    =====
    ===== экран*)

    (*-----
    ----- 0-я строка*)

        Write_Display(Single_mode,0,0,'ЖУРНАЛ АВАРИЙ');

    (*-----
    ----- 1-я и 2-я строки*)

        IF byAlarmsQty=0 THEN
            Write_Display(Single_mode,0,2,'пусто');
        ELSE

            Write_Display(Single_mode,0,1,funcDataTimeToString(dwAlarmTime[byCurr
ListPosition]));

            IF   trigAScreenIn.q   OR   tonScreenRefresh.q   OR
trigKeyPressed.q THEN

                IF byAlarmNumber[byCurrListPosition]<100 THEN

                    strAlarmDescription:=strAlarmText[byAlarmNumber[byCurrListPosition]];
                ELSE

```

```

    strAlarmDescription:=strAlarmTextCh[byAlarmNumber[byCurrListPosition]-
100];

    END_IF

    IF LEN(strAlarmDescription)>16 THEN
        IF bSpaceAhead THEN (*начать бегущую
строку с начала строки*)

            strLongLine:=CONCAT('
',strAlarmDescription);

        ELSE
            strLongLine:=strAlarmDescription;
        END_IF

        strTextLine:=MID(strLongLine,16,byFirstSymbol);

        tonRunLineDelay(in:=TRUE, pt:=t#500ms);
        IF      tonRunLineDelay.Q      THEN
byFirstSymbol:=byFirstSymbol+1; END_IF
            IF byFirstSymbol>LEN(strLongLine)THEN
                byFirstSymbol:=1;
                bSpaceAhead:=TRUE;
            END_IF
        ELSE
            strTextLine:=strAlarmDescription;
        END_IF
        strTextLine:=funcLineEndsFill(strTextLine,'right');
        Write_Display(Single_mode,0,2,strTextLine);

```

```

END_IF

(*-----
----- 3-я строка*)

strTextLine:=CONCAT('#',BYTE_TO_STRING(byCurrListPosition+1));
strTextLine:=CONCAT(strTextLine,'из');

strTextLine:=CONCAT(strTextLine,BYTE_TO_STRING(byAlarmsQty));
IF funcAlarmNewCheck(wAlarmNew,byCurrListPosition)
THEN strTextLine:=CONCAT('новое',strTextLine); END_IF

strTextLine:=funcLineEndsFill(strTextLine,'left');
IF byAlarmNumber[byCurrListPosition]<=2 THEN
strTextLine:=REPLACE(strTextLine,'$a0',1,1); END_IF (*индикация
критической аварии*)

Write_Display(Single_mode,0,3,strTextLine);
END_IF

(*=====
=====
===== нажатия*)

trigKeyPressed(clk:=wKeyPressed<>0);
IF trigKeyPressed.q AND NOT trigAScreenIn.q THEN

IF wKeyPressed=4 THEN (*сдвинуть список на
следующую запись*)

IF byCurrListPosition<byAlarmsQty-1 THEN
byCurrListPosition:=byCurrListPosition+1;

```

```

ELSE
    byCurrListPosition:=0;
END_IF
IF byAlarmsQty<>1 THEN prgRunLineIni(); END_IF

ELSIF wKeyPressed=8 THEN (*сдвинуть список на
предыдущую запись*)
    IF byCurrListPosition>0 THEN
        byCurrListPosition:=byCurrListPosition-1;
    ELSIF byAlarmsQty<>1 THEN
        byCurrListPosition:=byAlarmsQty-1;
    END_IF
    IF byAlarmsQty<>1 THEN prgRunLineIni(); END_IF

ELSIF wKeyPressed=64 THEN
    dintCurrWorkScreen:=GetWorkScreen();
    Switch_To_PREV_Mode();
END_IF
END_IF
END_IF
END_IF
END_IF
(*#####
#####
# журнал событий*)
    IF GetWorkScreen()=2 THEN
(*=====
=====
===== таймер возврата*)

```

```

tonScreenBack(in:=NOT tonScreenBack.q AND wKeyPressed=0,
pt:=t#30s);

```

```

IF tonScreenBack.q THEN

```

```

    dintCurrWorkScreen:=GetWorkScreen();

```

```

    Switch_To_PREV_Mode();

```

```

ELSE

```

```

(*=====
=====
===== инициализация*)

```

```

    tonScreenRefresh(in:=NOT tonScreenRefresh.q, pt:=t#200ms);

```

```

IF trigAScreenIn.q THEN

```

```

    byCurrListPosition:=0;

```

```

    Clear_Display(Single_mode);

```

```

    prgRunLineIni();

```

```

END_IF

```

```

(*=====
=====
===== экран*)

```

```

(*-----
----- 0-я строка*)

```

```

    Write_Display(Single_mode,0,0,'ЖУРНАЛ СОБЫТИЙ');

```

```

(*-----
----- 1-я и 2-я строки*)

```

```

IF byEventsQty=0 THEN

```

```

    Write_Display(Single_mode,0,2,'пусто');

```

```

ELSE

```

```
Write_Display(Single_mode,0,1,funcDataTimeToString(dwEventTime[byCurr  
ListPosition]));
```

```
IF trigAScreenIn.q OR tonScreenRefresh.q OR  
trigKeyPressed.q THEN
```

```
IF  
LEN(strEventText[byEventNumber[byCurrListPosition]])>16 THEN  
IF bSpaceAhead THEN (*начать бегущую  
строку с начала строки*)
```

```
strLongLine:=CONCAT(  
,strEventText[byEventNumber[byCurrListPosition]]);
```

```
ELSE
```

```
strLongLine:=strEventText[byEventNumber[byCurrListPosition]];
```

```
END_IF
```

```
strTextLine:=MID(strLongLine,16,byFirstSymbol);
```

```
tonRunLineDelay(in:=TRUE, pt:=t#500ms);
```

```
IF tonRunLineDelay.Q THEN
```

```
byFirstSymbol:=byFirstSymbol+1; END_IF
```

```
IF byFirstSymbol>LEN(strLongLine)THEN
```

```
byFirstSymbol:=1;
```

```
bSpaceAhead:=TRUE;
```

```
END_IF
```

```
ELSE
```



```

strTextLine:=strEventText[byEventNumber[byCurrListPosition]];
                                END_IF
                                strTextLine:=funcLineEndsFill(strTextLine,'right');
                                Write_Display(Single_mode,0,2,strTextLine);
                                END_IF

(*-----
----- 3-я строка*)

strTextLine:=CONCAT('#',BYTE_TO_STRING(byCurrListPosition+1));
strTextLine:=CONCAT(strTextLine,'из');

strTextLine:=CONCAT(strTextLine,BYTE_TO_STRING(byEventsQty));
strTextLine:=funcLineEndsFill(strTextLine,'left');

                                Write_Display(Single_mode,0,3,strTextLine);
                                END_IF

(*=====
=====
===== нажатия*)

                                trigKeyPressed(clk:=wKeyPressed<>0);
                                IF trigKeyPressed.q AND NOT trigAScreenIn.q THEN

                                        IF wKeyPressed=4 THEN (*сдвинуть список на
следующую запись*)

                                                IF byCurrListPosition<byEventsQty-1 THEN
                                                        byCurrListPosition:=byCurrListPosition+1;
                                                ELSE

```

```

        byCurrListPosition:=0;
    END_IF
    IF byEventsQty<>1 THEN prgRunLineIni(); END_IF

    ELSIF wKeyPressed=8 THEN (*сдвинуть список на
предыдущую запись*)
        IF byCurrListPosition>0 THEN
            byCurrListPosition:=byCurrListPosition-1;
        ELSIF byEventsQty<>1 THEN
            byCurrListPosition:=byEventsQty-1;
        END_IF
        IF byEventsQty<>1 THEN prgRunLineIni(); END_IF

    ELSIF wKeyPressed=64 THEN
        dintCurrWorkScreen:=GetWorkScreen();
        Switch_To_PREV_Mode();
    END_IF
END_IF
END_IF
END_IF
END_IF

END_IF
(*#####
#####
#### индикаторы*)
(*-----
----- индикаторы контроллера*)
IF byWorkMode=255 THEN

```

```

tonIndikatorRun(in:=NOT tonIndikatorRun.q, pt:=T#50ms);
IF tonIndikatorRun.q THEN
    byIndikator:=0;
    byI:=SEL(byI<5,0,byI+1);
    CASE byI OF
        0:byIndikator.0:=TRUE;
        1:byIndikator.1:=TRUE;
        2:byIndikator.2:=TRUE;
        3:byIndikator.3:=TRUE;
        4:byIndikator.4:=TRUE;
        5:byIndikator.5:=TRUE;
    END_CASE
END_IF
ELSIF bAlarmCaution AND NOT bSystemRun THEN
    byIndikator:=63;
ELSIF bAlarmCaution THEN
    BlinkIndikator(enable:=TRUE, timelow:=T#600ms,timehigh:=T#600ms);
    IF BlinkIndikator.OUT THEN
        byIndikator:=63;
    ELSE
        byIndikator:=0;
    END_IF
ELSIF bSystemRun THEN
    tonIndikatorRun(in:=NOT tonIndikatorRun.q, pt:=T#200ms);
    IF tonIndikatorRun.q THEN
        byIndikator:=0;
        byI:=SEL(byI>0,5,byI-1);
        CASE byI OF

```

```

        0:byIndikator.0:=TRUE;
        1:byIndikator.1:=TRUE;
        2:byIndikator.2:=TRUE;
        3:byIndikator.3:=TRUE;
        4:byIndikator.4:=TRUE;
        5:byIndikator.5:=TRUE;

    END_CASE

END_IF

ELSE

    byIndikator:=0;

END_IF

(*-----
----- лампа на лицевой панели*)

IF bAlarmCaution AND NOT bSystemRun AND byWorkMode<>255 THEN
    bCabRedLight:=1;
ELSIF bAlarmCaution THEN
    BlinkCabRedLight(enable:=TRUE, timelow:=T#600ms,timehigh:=T#600ms);
    IF BlinkCabRedLight.OUT THEN
        bCabRedLight:=1;
    ELSE
        bCabRedLight:=0;
    END_IF
ELSE
    bCabRedLight:=0;
END_IF

```